TECHNICAL UNIVERSITY OF MUNICH
TUM School of Engineering and Design

Change detection for monitoring of man-made objects using time series of very high resolution spaceborne SAR images

Carlos Villamil Lopez

Dissertation

2023

TECHNISCHE UNIVERSITÄT MÜNCHEN
TUM School of Engineering and Design

Change detection for monitoring of man-made objects using time series of very high resolution spaceborne SAR images

Carlos Villamil Lopez

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr. rer. nat. Niklas Boers
Prüfer der Dissertation:	1. Prof. Dr. rer. nat. Martin Werner
	2. Prof. Dr.-Ing. habil. Michael Schmitt
	3. Prof. Dr.-Ing. Alberto Moreira

Die Dissertation wurde am 24.02.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 07.07.2023 angenommen.

Abstract

Earth observation (EO) satellites with very high resolution (VHR) capabilities can provide regular and repeatable observations of many locations across the globe over long time periods, and are a cost-effective way to monitor remote and inaccessible areas. These satellites can be exploited to monitor different types of human activity, delivering insights which can be used to inform policy- and decision-making. Synthetic aperture radar (SAR) sensors are especially interesting because they can provide VHR images independently of sunlight and under all weather conditions, and are also very well suited for change detection. This thesis presents three novel methods for the monitoring of man-made objects using time series of VHR SAR images, which are then applied to address different practical applications.

First, a method for the automatic estimation of all the relevant parameters of oil storage tanks is introduced, which can be applied for the monitoring of oil inventories. The dimensions of each storage tank, as well as the fill level in the case of tanks with a floating roof, are derived from its semicircular double reflections, which are detected using the coherent scatterers (CSs) in the SAR image. When a time series is available, the temporal information is exploited to provide more accurate and robust estimates. This method is application specific, but illustrates the monitoring of well-known, static objects, showing how prior knowledge on an object's geometry and approximate location can be exploited.

Secondly, an unsupervised change detection (CD) method is introduced. Changes are detected by the appearance and disappearance of CSs. These CSs are detected in each image and compared coherently across an image pair or a time series. This enables the detection of changes caused by the appearance, disappearance, or movement of man-made objects, as well as modifications to static objects, while ignoring changes to natural targets such as vegetation. The detected changes can be categorized according to their temporal behavior, and the corresponding changed objects can be segmented. As an example on the monitoring of unknown static objects in complex scenes, this method is applied to monitor construction activity in urban areas.

Lastly, a method for the detection and classification of objects in individual SAR images is presented. This task is formulated as a template matching problem, and a fully convolutional Siamese network architecture is used. This approach is based on supervised learning but can be trained with relatively few labelled samples, and also takes into account the specific characteristics of SAR images, such as the significant effect that the imaging geometry has on an object's appearance. As an example of the monitoring of moving objects, this method is applied to detect and classify different types of airplanes parked at airports. When applied to a time series in combination with the CD method, the arrival and departure dates of the detected airplanes can also be estimated, enabling a detailed monitoring of the level of activity at an airport.

The proposed methods have been evaluated using time series of TerraSAR-X images with submeter resolution. The analysis of the obtained results has shown that all three methods perform well, indicating that they could be used in practice for the intended applications. While the method for the monitoring of oil storage is application specific, the methods for change detection and object recognition are more general and could be applied to monitor other types of man-made objects. Besides, as shown in this thesis, these two methods complement each other.

Kurzfassung

Erdbeobachtungssatelliten mit hohem räumlichen Auflösungsvermögen sind in der Lage Bildmaterial global mit regelmäßiger Überdeckung zu liefern. Auf kostengünstige Weise können entlegene und schwer zugängliche Gebiete überwacht werden. Ein mögliches Einsatzszenario ist das Monitoring menschlicher Aktivitäten, um Erkenntnisse als Grundlage für politische Entscheidungen zu erlangen. In diesem Fall ist der Einsatz von Plattformen mit Synthetischem Apertur Radar (SAR) von besonderem Interesse, da sie entfernungs-, tageszeit- und wetterunabhängig Aufnahmen mit einem hohen Auflösungsvermögen ermöglichen. Damit ist ein SAR-Sensor ideal zur Detektion von Änderungen am Boden geeignet. In der vorliegenden Dissertation wurden drei neuartige Methoden entwickelt, welche menschliche Aktivitäten aus SAR-Zeitserien detektieren und höherwertige Informationen extrahieren.

Mit der ersten Methode ist es möglich, sämtliche relevanten Parameter von Öltanks automatisch zu schätzen, um beispielsweise Bestände von Öl-Reservoirs gesamtheitlich zu bestimmen. Die geometrischen Maße einzelner Tanks bzw. deren Füllstände im Falle von Schwimmdeckeltanks können mit dem vorgestellten Verfahren aus ihren halbkreisförmigen Signaturen unter Berücksichtigung sogenannter kohärenter Streuzentren, auf äußerst robuste Weise abgeleitet werden. Ist zusätzlich eine interferometrische Zeitserie verfügbar, können die Schätzungen aus den Einzelbildauswertungen in ihrer Genauigkeit und Robustheit verbessert werden. Das entwickelte Verfahren illustriert die Möglichkeit eines Monitoring von statischen Objekten, deren Positionen a priori bekannt sind, um höherwertige Parameter der Objekte automatisiert zu extrahieren.

Die zweite Methode verwendet ebenfalls kohärente Streuzentren (CSs), jedoch um Änderungsdetektion in wesentlich allgemeinerer Form zu betreiben. Im Speziellen werden die CSs in den Einzelbildern erkannt und zusätzlich ihre Kohärenz entlang einer interferometrischen Zeitreihe ausgewertet, um das Auftauchen, Verschwinden, Verändern oder die Bewegung von künstlichen Objekten zu detektieren, wogegen Änderungen an natürlichen Zielen (z.B. Vegetation) nicht berücksichtigt werden. Die erkannten Veränderungen können nach ihrem chronologischen Verhalten kategorisiert und segmentiert werden. Als praxisnahe Anwendung wird diese Methode zum Monitoring von Bauaktivitäten in städtischen Gebieten demonstriert.

Abschließend wird eine Methode zur Detektion und Klassifikation von Objekten in Einzelbilddaten vorgestellt. Die Aufgabe wird als sogenanntes Template-Matching Problem formuliert und als Lösungsansatz tiefes Lernen auf Basis einer Siamesischen Netzwerkarchitektur eingesetzt. Da es sich um überwachtes Lernen handelt, mussten annotierte Trainingsdaten erzeugt werden. Das neuartige Verfahren zeichnet sich durch eine sehr gute Erkennungsleistung unter Verwendung einer geringen Zahl an Trainingsdaten aus. Spezifische Signatureigenschaften wie z.B. der Einfluss von Aufnahmegeometrie oder wechselnde Bodenverhältnisse können berücksichtigt werden. Als anschauliches praxisnahes Experiment wurde diese Methode zur Erkennung von Flugzeugsignaturen eingesetzt. Wird zusätzlich zur Klassifikation einzelner Aufnahmen eine Zeitreihenauswertung durchgeführt, können beispielsweise auch die Ankunfts- und Abflugzeiten der Flugzeuge geschätzt werden, um Aktivitätsanalysen eines Flughafens zu ermöglichen.

Sämtliche Methoden wurden auf Bildmaterial der TerraSAR-X Mission (Auflösung kleiner ein Meter) erfolgreich angewandt. Die in der Arbeit gezeigten Ergebnisse unterstreichen die Praxistauglichkeit der selbst entwickelten Verfahrensweisen. Die Methode zum Monitoring von Öltanks bietet Lösungen für eine sehr spezifische Anwendung. Die beiden anderen Methoden zur Änderungsdetektion bzw. zur Signaturklassifikation lassen sich auf weitere Objekttypen erweitern. Zusätzlich bieten sich die beiden Methoden für eine kombinierte Anwendung an.

Contents

List of Abbreviations

Abbreviation	Description	Page
AP	Average precision	117
ATI	Along-track interferometry	20
ATR	Automatic target recognition	21
CCD	Coherent change detection	22
CD	Change detection	22
CFAR	Constant false alarm rate	32
CNN	Convolutional neural network	30
CPU	Central processing unit	127
CS	Coherent scatterer	24
dB	Decibel	39
DEM	Digital elevation model	20
DInSAR	Differential synthetic aperture radar interferometry	20
ECEF	Earth-centered, Earth-fixed	44
EO	Earth observation	15
FFT	Fast Fourier transform	40
GPU	Graphics processing unit	80
GCPs	Ground control points	21
GMTI	Ground moving target indicator	20
ICD	Incoherent change detection	22
InSAR	Synthetic aperture radar interferometry	20
IoU	Intersection over union	87
LEO	Low Earth orbit	17
NMS	Non-maximum suppression	87
NN	Neural network	32
OSM	OpenStreetMap	34
PSI	Persistent scatterer interferometry	20
RAM	Random-access memory	127
RPC	Rational polynomial coefficients	44
SAR	Synthetic aperture radar	16
SDGs	Sustainable Development Goals	15
SGD	Stochastic gradient descent	112
SiamFC	Siamese fully convolutional	77
SLC	Single look complex	37
SVM	support vector machine	58
TomoSAR	SAR tomography	20
UN	United Nations	15
UTM	Universal transverse mercator	91
VHR	Very high resolution	15
WGS84	World geodetic system 1984	44

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Multiple global challenges need to be addressed in the next decades, such as those outlined in the United Nations (UN) 2030 Agenda for Sustainable Development and its 17 Sustainable Development Goals (SDGs) [United Nations, 2015]. To achieve these goals, it is fundamental to monitor and document progress using a data-driven and evidence based approach, and use this knowledge to inform policymaking. Earth observation (EO) can provide accurate and robust data on a continuous basis, and is already widely used to measure many of the progress indicators of the SDGs [European Space Agency, 2020; Persello et al., 2022]. While EO data can be acquired using different types of sensors, spaceborne sensors are better suited for addressing such global challenges. In contrast to in-situ sensors, drones or airplanes, satellites provide a synoptic view of the Earth surface, enable regular and repeatable observations over long time periods, and are a cost-effective way to monitor remote and inaccessible areas [European Space Agency, 2020].

EO data with medium resolution (i.e., around ten meters per pixel) from the Landsat [Wulder et al., 2019] and Copernicus [Aschbacher, 2017] programs can be used to address many of these global challenges. This kind of satellite missions provide a global coverage every few days, and their data is available free of charge thanks to their open data policies. However, certain applications either require or greatly benefit from the availability of very high resolution (VHR) images (i.e., a few decimeters per pixel). The satellites acquiring this high resolution data cannot regularly provide global observations, but they are very well suited for the frequent monitoring of many specific locations across the globe. Until recently, access to VHR data was limited, with these images either restricted to a few national agencies or sold at high prices by a few commercial providers. However, in the last decade, NewSpace companies have achieved a reduction in the costs of developing, manufacturing and launching satellites, which has led to many companies launching or planning to launch large satellite constellations [Kulu, 2021]. Some of these are constellations of EO satellites with VHR capabilities, which will increase the availability of such data and is also expected to make it cheaper and easier to access. This is posed to enable applications that until recently were either not feasible or cost prohibitive.

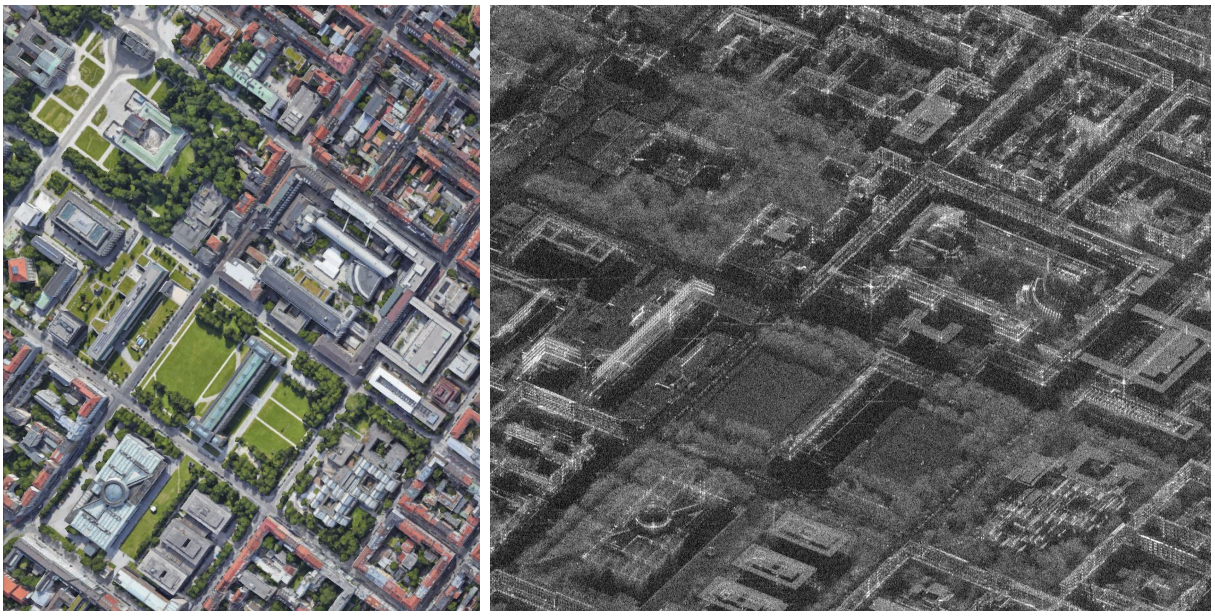
Among the applications that greatly benefit from the availability of VHR data are those involving the frequent observation of human activity and man-made objects. As with the SDGs, the information obtained from monitoring different kinds of human activity from space can and should also be used to inform policy- and decision-making. In the last couple of years alone, the world has faced several political and humanitarian crises, an energy crisis, a global pandemic, disruptions to the global supply chain, multiple wars and armed conflicts, etc. These events have greatly affected the global economy and the well-being and security of millions of people. Some of the causes and/or consequences of this kind of events can be observed in VHR satellite imagery. For example, some of the effects of the COVID-19 pandemic to society and the economy have been quantified using EO data [Anghelea et al., 2021; Falkowski et al., 2021; Hamamoto et al., 2021; Wu et al., 2021]. EO data is also used to analyze energy production (e.g., by

monitoring global oil inventories [Orbital Insight, nd] or the deployment of solar panels [Yu et al., 2018]). VHR satellite images are also often used to provide transparency to humanitarian crises or human right violations happening in remote locations, like modern slavery [Boyd et al., 2018] or forced migrations [Sprohnle et al., 2017; Ghorbanzadeh et al., 2022]. Finally, governments and other international institutions also use this kind of imagery to assess the effects of war [United Nations Satellite Centre, 2016, 2022] or to monitor the security of critical infrastructure [Lafitte & Robin, 2015]. What most of these applications of VHR satellite imagery have in common, is that they involve the monitoring of certain changes associated to specific types of man-made objects (e.g., alterations to buildings or infrastructure, movement of vehicles...). Traditionally, this type of analysis has often involved at least some degree of manual work by human image analysts. However, with the ever increasing amount of EO data, it is fundamental to develop methods that can automatically extract information from the available imagery, so that actionable insights can be directly delivered to the different stakeholders.

Two types of sensors are typically used for acquiring VHR satellite imagery: optical and synthetic aperture radar (SAR) sensors. Two images acquired by these two types of sensors showing the area around the city campus of the Technical University of Munich (TUM) can be seen in Fig. 1.1. Traditionally, optical imagery has been more widely used, as it is generally considered to be easier to interpret. However, SAR sensors have several specific advantages, and the number of SAR satellites with VHR capabilities is increasing rapidly.

1.2 Spaceborne synthetic aperture radar

A synthetic aperture radar (SAR) is an active imaging sensor capable of providing high resolution two-dimensional images independently of sunlight and under all weather conditions [Moreira et al., 2013]. The SAR imaging principle is based on an invention by Wiley [1954], which was later extended to the principle of the synthetic aperture as it is known today [Cutrona et al., 1961;



(a) (b)

Figure 1.1: Optical and SAR satellite images of the city campus of the Technical University of Munich. a) Optical image (obtained from Google Earth). b) SAR image (acquired with TerraSAR-X). Both images depict approximately the same area but were acquired from different perspectives and at different times.

Willey, 1985]. A SAR system comprises a side-looking radar mounted on a forward moving platform. As the platform moves, the radar system sequentially transmits electromagnetic waves with high power and receives the backscattered echoes. The geometrical and dielectric properties of the imaged objects will determine which fraction of the transmitted energy gets backscattered to the radar, and therefore the amplitude and phase of the received signal. The backscattered signals received at the different positions can be then added coherently during processing to form a two-dimensional image depicting the reflectivity of the imaged region. This processing technique allows to achieve a high resolution in azimuth (i.e., the direction of movement), which would otherwise require using an impractically large antenna. Using this approach, the resolution achieved both in range (i.e., the direction of the radar's line of sight) and azimuth is independent of the distance to the imaged scene, which is fundamental for high resolution imaging from space.

The feasibility of spaceborne SAR was first demonstrated in 1964 with the launch of the experimental Quill satellite, a proof-of-concept mission by the National Reconnaissance Office of the United States [Butterworth, 2004], which was only recently declassified. Satellites equipped with SAR sensors have been widely used for Earth remote sensing since 1978, when the first civilian SAR satellite (Seasat) was launched [Jordan, 1980]. They enable reliable and continuous monitoring of dynamic processes on the Earth surface, as no other spaceborne sensors are capable of acquiring high resolution images day-and-night and even during adverse atmospheric conditions (e.g., cloud coverage, smog, etc.). However, SAR images are often considered more difficult to interpret than optical images, as they are affected by radar specific imaging effects such as speckle and geometric distortions like layover and foreshortening. The speckle effect, which causes SAR images to appear noisy, is a result of the coherent sum of the contributions from many elemental scatterers which are randomly distributed inside each resolution cell [Moreira et al., 2013]. Speckle appears as multiplicative noise and can therefore not be reduced by increasing the radar's transmit power. The geometric distortions are caused by the range-based imaging principle and the fact that SAR images are a projection of a three-dimensional scene into the radar's coordinate system (i.e., range and azimuth).

Spaceborne SAR data is used for many different applications, such as environmental monitoring and climate change research, hazard and disaster monitoring, 3-D and 4-D (space and time) mapping, change detection, security-related applications and even planetary exploration [Moreira et al., 2013]. Different frequency bands are better suited for different applications. The lower the frequency, the deeper that the radar signal can penetrate into different media (e.g., vegetation, ice and snow, dry soil). However, using higher frequency bands makes it easier to achieve a higher resolution. Typically, spaceborne SAR missions operate in either L, S, C or X band, as lower frequencies would require a very large antenna, and at higher frequencies the atmospheric attenuation increases for high humidity [Altshuler & Marr, 1988]. In contrast, atmospheric attenuation does not play such an important role for airborne SAR sensors, enabling them to operate at much higher frequencies like W band [Palm, 2021]. In addition to using different frequencies, the use of different polarizations for transmitting and/or receiving the radar signal can also provide additional information on the scattering properties of the imaged targets. Some advanced SAR sensors can acquire data using multiple polarizations simultaneously, which allows to identify different scattering mechanisms by applying techniques such as SAR polarimetry [Hajnsek & Desnos, 2021].

SAR satellites are typically placed on low Earth orbit (LEO), as higher orbits introduce some additional challenges (e.g., power budget) [Matar et al., 2020]. LEO orbits typically result in a repeat-pass interval in the order of 10 to 20 days. This is the time interval required so that a single satellite can image a given location twice with the same imaging geometry. When considering acquisitions with different imaging geometries (e.g., different incidence angles, ascending

and descending orbits, left and right looking), the time interval between the acquisition of two images of the same location is greatly reduced. However, SAR time series with repeat-pass acquisitions are especially interesting, as they are better suited for change detection, and they also enable coherent analysis (i.e., by comparing the phase of two or more complex-valued SAR images). This is exploited by techniques like SAR interferometry, which can be used to accurately measure surface topography or ground deformation [Moreira et al., 2013]. In order to reduce the repeat-pass interval, constellations with multiple identical satellites can be used. Alternatively, interferometric measurements can also be performed in a single-pass by using two satellites flying in close formation and operating in a bistatic configuration [Zink et al., 2021].

Current spaceborne SAR sensors enable flexible image acquisitions, as they are capable of operating in different imaging modes by steering the antenna, which can be done either mechanically or electronically. These different modes provide a trade-off between the size of the imaged area and the achievable resolution. The most basic mode, named Stripmap, simply images a continuous strip and does not require to steer the antenna. In modes like ScanSAR or TOPSAR, the antenna is steered to image a larger area, but this results in a coarser resolution. On the other hand, the Spotlight mode steers the antenna to illuminate a single spot during a longer time period, which results in a higher resolution. As an example, the TerraSAR-X satellite [Buckreuss et al., 2009] can image an area of 4×3.7 km with a resolution of 0.6×0.25 m in the Staring Spotlight mode [Mittermayer et al., 2014], and an area of approx. 270×1500 km with a resolution of approx. 2×40 m in Wide ScanSAR mode [Steinbrecher et al., 2014].

1.2.1 Current and planned spaceborne SAR missions

In order to get a better understanding of current trends in the SAR industry and assess the future availability of spaceborne SAR data, it is fundamental to look into the currently active spaceborne SAR missions, as well as the missions planned for the next few years. An overview of these missions is provided below in Table 1.1. Each row in this table lists the name of the corresponding SAR mission, the involved countries, the used frequency band (or bands, for a few missions with multi-frequency capabilities), the number of satellites and the years they were launched are listed. The table is sorted according to the year in which the first satellite of each mission was launched. In case that some of the satellites in a constellation have not been launched yet, both the current number of active satellites and the planned number are listed. Also, for conciseness, older missions that stopped operating several years ago are not listed in this table. Finally, some of the classified reconnaissance SAR satellite missions (e.g., Russia's Condor [Vladimir et al., 2016] or China's Yaogan [Deng et al., 2019] satellites) are not listed due to the inherent lack of transparency and difficulty of finding reliable information.

Spaceborne SAR images with a resolution of 1 meter or better have been publicly available since 2007, with the launch of satellites like TerraSAR-X and TanDEM-X [Buckreuss et al., 2009; Krieger et al., 2007] and COSMO-SkyMed [Caltagirone et al., 2014], which are listed towards the top of Table 1.1. These missions are used for both scientific and commercial purposes and, in the case of COSMO-Skymed, also military, as it is a dual-use system. Many years later, these satellites are still operative and delivering high quality imagery [Zink et al., 2021; Calabrese et al., 2018]. Typically, this kind of missions with high resolution capabilities operate in X-band, as higher frequencies make it easier to achieve a higher bandwidth and therefore a better range resolution. Complementary to these missions, satellites like Sentinel-1 [Snoeij et al., 2010] or ALOS-2 [Kankaku et al., 2013] operate in lower frequencies like C- and L-band, and aim to provide global coverage with mid-resolution. Interestingly, the majority of the the missions listed in Table 1.1 operate in X-band and have high resolution capabilities.

Table 1.1: Current and planned spaceborne SAR missions.*

Mission	Countries	Freq. bands	Launch years	Satellites
SAR-Lupe	Germany	X	2006 - 2008	5
TanDEM-X	Germany	X	2007 - 2010	2
COSMO-SkyMed	Italy	X	2007 - 2010	4
Radarsat-2	Canada	C	2007	1
TecSAR 1, 2	Israel	X	2008 - 2014	2
RISAT-2	India	X	2009	1
KOMPSAT-5	South Korea	X	2013	1
ALOS-2	Japan	L	2014	1
Sentinel-1	Europe (ESA)	C	2014 - 2024	2/4
Gaofen-3	China	C	2016 - 2022	3
Paz	Spain	X	2018	1
NovaSAR	UK	S	2018	1
ASNARO-2	Japan	X	2018	1
SAOCOM	Argentina	L	2018-2020	2
RCM	Canada	C	2019	3
CSG	Italy	X	2019 -	2/4
Gaofen 10R	China	?	2019	1
Gaofen-12	China	?	2019 - 2022	3
ICEYE	Finland	X	2018 -	24/?
Spacety	China	C, X	2020 -	2/?
Capella	USA	X	2018 -	8/?
SARah	Germany	X	2022 -	1/3
Ludi Tance 1	China	L	2022	2
Umbra	USA	X	2021 -	4/24
Synspective	Japan	X	2020 -	3/30
QPS-SAR	Japan	X	2019 -	2/36
PredaSAR	USA	X	2023 -	0/48
ALOS-4	Japan	L	2023	0/1
KOMPSAT-6	South Korea	X	2023	0/1
Biomass	Europe (ESA)	P	2024	0/1
NISAR	India, USA	S, L	2024	0/1
ROSE-L	Europe (ESA)	L	2028	0/1

Another interesting trend is that, while most of the earlier missions are mainly operated by national space agencies, many of the newer ones are completely private initiatives by companies which design, build, launch and operate the satellites. With the exception of military reconnaissance missions, most of the new and future publicly funded SAR missions operate on lower frequency bands (P, L, S and C), have polarimetric capabilities and are focused on environmental monitoring. On the other hand, the new private SAR missions use sensors operating in X-band, with a single polarization and VHR capabilities (i.e., sub-meter resolution), and have a focus on commercial applications. The companies launching these missions follow a NewSpace approach, and aim to lower the overall costs by reducing the satellite's weight, using off-the-shelf components whenever possible and eliminating redundancy. While this approach increases the satellite's failure rate and leads to satellites with slightly reduced capabilities, the significant lower costs make it feasible to build large constellations with dozens of satellites. This can be seen in Table

*Last updated on February 5th, 2023.

1.1, with several future missions planning to launch between 24 to 48 satellites. Additionally, current missions like ICEYE [Muff et al., 2022] and Capella [Stringham et al., 2019] already have many working satellites in orbit and plan to keep increasing this number. These missions are already enabling very frequent VHR SAR observations of any given location, and they will soon likely achieve a repeat-pass time in the order of several hours. Besides, they are also making it both easier and cheaper to access to VHR data. The current situation, with a rapidly increasing number of SAR satellites in orbit and more SAR data available than ever before, is often referred to as the "Golden Age of SAR" [Moreira, 2014; Zhang & Lu, 2022]. These current trends have resulted in an increased demand for algorithms capable of automatically extracting information from all this newly available VHR SAR data.

1.3 Applications of very high resolution SAR data

The brief analysis of the current and planned SAR missions has shown a clear trend: a rapidly increasing number of SAR satellites operating in X-band, with a single polarization and VHR capabilities. In this section, an overview of some of the most popular and interesting applications for such data will be provided.

1.3.1 Measurement of physical quantities

Modern SAR satellites like TerraSAR-X can be used to perform very accurate measurements: ranging accuracy in the centimeter level can be achieved by accounting for certain error sources like solid Earth tides and tropospheric water vapor [Eineder et al., 2011; Cong et al., 2012]. Even though SAR sensors acquire two-dimensional images, these can be exploited for performing accurate 3-D and 4-D (space and time) mapping [Moreira et al., 2013]. Using techniques such as across-track SAR interferometry (InSAR) [Bamler & Hartl, 1998] or radargrammetry [Crosetto & Pérez Aragüés, 1999], two SAR images of the same location can be processed to generate a digital elevation model (DEM), effectively measuring the topography of the imaged area. Advanced InSAR techniques can also exploit more than two SAR images. For example, differential SAR interferometry (DInSAR) techniques using three or more SAR images can be applied to measure ground deformation, enabling the detection of displacements of the Earth surface at a wavelength scale [Moreira et al., 2013]. When long time series with many SAR images are available, persistent scatterer interferometry (PSI) [Ferretti et al., 2001] can be applied to generate accurate 3-D point clouds composed by point scatterers that remain stable throughout the observation period, and for which the deformation can also be precisely estimated. Additionally, multiple baselines can also exploited by applying SAR tomography (TomoSAR) [Reigber et al., 1999; Zhu, 2011] techniques to achieve actual 3-D imaging. Instead of simply estimating a single height value for each pixel, TomoSAR methods are able to separate different scatterers inside a single image pixel, which are located at different heights but appear in the same pixel due to layover.

In addition to measuring heights and deformation, SAR sensors can also be used to measure velocities: ground moving target indication (GMTI) methods exploiting along-track interferometry (ATI) can detect moving targets (e.g., cars or ships) and estimate their velocities [Meyer et al., 2006; Makhoul et al., 2015; Baumgartner & Krieger, 2016]. These methods have been applied for the monitoring of car traffic flows using TerraSAR-X data [Suchandt et al., 2010]. Additionally, ATI can also be applied to measure current fields in the open ocean, coastal waters, and rivers [Romeiser et al., 2005, 2010].

Even though most of these techniques have been applied for many years and do not necessarily require VHR SAR data, it has been shown that a higher spatial resolution provides some important advantages. For example, urban details such as building heights and deformation can be detected

in interferograms and differential interferograms generated with high resolution SAR data [Eineder et al., 2009]. In contrast, when using medium resolution data, this information can only be obtained for isolated points by applying PSI approaches, which require an increased number of images. Additionally, an improved spatial resolution results in a significant increase in the density of the point clouds obtained when applying PSI or TomoSAR, as well as in more accurate height estimates [Ge et al., 2018]. This enables the detection of very localized deformation patterns even on different parts of a single building [Zhu et al., 2018]. By combining all these capabilities with the aforementioned ranging accuracy, modern SAR sensors with VHR capabilities can be exploited to measure the absolute 3-D position of targets with very high accuracy [Gisinger et al., 2015; Montazeri, 2019]. This has led to the use of spaceborne SAR data for the definition of ground control points (GCPs) [Montazeri et al., 2018].

SAR interferometry is a well researched topic, and these different approaches are often used to process spaceborne SAR data, enabling accurate measurements of the complete imaged scenes. However, they require at least two images, which typically cannot be acquired in a single satellite pass. The only exception to this are SAR missions capable of performing bistatic acquisitions, such as TanDEM-X [Zink et al., 2021]. This means that with most spaceborne SAR sensors, only the objects that remain static (or exhibiting only small deformations) between these two or more image acquisitions can be measured using these techniques. Alternatively, the dimensions of a given object can also be measured using a single SAR image, as long as a few relevant points can be identified in the image. In this case, heights can be estimated by exploiting the layover or shadow effects and accounting for the imaging geometry. The accuracy achievable using this approach mainly depends on the spatial resolution of the sensor (which for VHR SAR sensors can be in the order of a few decimeters) and the imaging geometry (e.g., incidence angle). Due to the need to identify some relevant points of the objects to be measured, this measurement approach is typically either applied manually by an human image analyst, or requires the development of methods for measuring specific types of objects. For example, this measurement principle has been applied to estimate the amount of oil inside storage tanks [Guida et al., 2010; Hammer et al., 2017] or to measure building heights [Sun et al., 2022c].

1.3.2 Object detection and classification

Traditionally, the task of detecting and classifying objects in SAR images has been referred to as automatic target recognition (ATR) [Dudgeon & Lacoss, 1993]. Historically, SAR ATR methods have been developed with a focus on security related applications, with most methods being evaluated using the MSTAR dataset [Ross et al., 1998]. This dataset includes a large number of image chips of VHR X-band SAR data with many different imaging geometries for 10 different classes of military vehicles.

In recent years, the interest on object detection and classification in SAR images has increased due to the rapidly growing number of SAR satellites with VHR capabilities. Also, the great success of deep learning methods for the analogous task in optical images [Zhu et al., 2017; Guo et al., 2018] and their subsequent application to SAR images has enabled significant progress in the field of SAR ATR [Zhu et al., 2021]. When adapting deep learning methods originally developed for optical images, it is fundamental to take into account the specific characteristics of SAR data: such as its very large dynamic range, the different statistics and multiplicative speckle noise, as well as the range and azimuth based imaging and its associated effects such as shadow and layover [Zhu et al., 2021]. In addition to this, the side-looking imaging geometry of SAR sensors implies that the appearance of a given object will vary significantly depending on its orientation with respect to the radar sensor. This makes some popular data augmentation techniques which are often applied to nadir optical imagery (e.g., rotating an object) not applicable to SAR data. All

this, coupled with the lack of good quality standardized datasets, makes the use of deep learning for object detection and classification in SAR imagery somewhat more challenging than for the case of nadir optical imagery. Nevertheless, unlike optical sensors, SAR sensors can be used to perform object detection and classification day-and-night and during all-weather conditions, and deep learning has also been applied very successfully to VHR SAR data. Nowadays, the MSTAR dataset is considered easy, and new research work, which will be reviewed later in Section 2.3, is focused on newer datasets with different object types.

1.3.3 Change detection

Spaceborne SAR sensors are especially well suited to monitor changes on the ground. SAR is an active sensor which provides its own illumination, and can therefore ensure a consistent illumination of the same scene at different times. Therefore, when comparing two SAR images acquired with the same sensor and imaging geometry (i.e., using a repeat-pass orbit) at different times, any significant differences between them will be due to changes in the imaged scene. SAR missions with mid-resolution sensors provide a global coverage every few days and can be exploited to monitor changes of large or moderate size on a regional or global scale. On the other hand, missions with high resolution capabilities can regularly acquire much more detailed images of specific locations, enabling the detection of smaller changes. Change detection (CD) techniques exploiting these high resolution SAR images can be applied to monitor man-made objects. Changes caused by their appearance, disappearance, or movement inside the imaged scene can be detected, as well as changes to static objects. This enables the monitoring of different types of human activity, such as the arrival and departure of airplanes at airports [Villamil Lopez & Stilla, 2018] and of ships at ports [Villamil Lopez et al., 2017], the construction of new buildings [Villamil Lopez & Stilla, 2022; Marin et al., 2015], or the movement of shipping containers and parked cars [Bovolo et al., 2013].

CD methods with SAR images can be classified into coherent and incoherent [Preiss et al., 2006; Rignot & van Zyl, 1993]. Incoherent change detection (ICD) methods detect changes by comparing the amplitude of two co-registered SAR images, while coherent change detection (CCD) methods detect changes by the loss of coherence. CCD methods can detect subtle changes [Preiss et al., 2006], such as those caused by vehicles when driven over soft surfaces [Cha et al., 2015], or by objects when displaced to a distance smaller than the spatial resolution. Those changes would not typically be resolved with the amplitude of the SAR images. However, CCD requires a short time interval between the image acquisitions (i.e., a short temporal baseline), as otherwise temporal decorrelation will be significant [Reigber et al., 2016], leading to the vegetation and other natural targets in the scene to be detected as changes even if no significant change occurred. This is especially relevant at X-band, which is used by most SAR satellites with VHR capabilities, as the decorrelation rate increases at higher frequencies [Parizzi et al., 2009]. On the other hand, while less sensitive to subtle changes, ICD methods are more flexible as they can be applied to image pairs with longer temporal baselines, and in some cases even to image pairs with slightly different imaging geometries (e.g., incidence angles) as shown in [Tao & Auer, 2016].

1.4 Objectives and contributions

So far, the importance of monitoring human activity from space has been introduced, and how the rapidly growing number of SAR satellites with VHR capabilities are well suited for this task. As outlined in Section 1.3, the data acquired by these sensors can be used to perform accurate measurements, identify specific types of objects and detect changes. Different types of human activity can be monitored by using VHR SAR data to detect certain changes associated to specific

types of man-made objects. This thesis will focus on the development of novel methods for the monitoring of man-made objects using time series of VHR SAR images. These methods will be then applied to address several practical applications.

The proposed methods will attempt to exploit any available prior knowledge of the imaged scenes and the objects to be monitored. The following three cases will be distinguished and treated in this thesis, which correspond to varying amounts of prior knowledge available:

- Monitoring of well-known, static objects. In this case, there is some prior knowledge of the geometry and location of the objects to be monitored. This is often the case for critical infrastructure.
- Monitoring of unknown static objects in complex scenes. In this case, accurate and up-to-date prior knowledge of the imaged scene cannot be guaranteed. However, the objects of interest can be observed over long time periods, and their temporal behavior can be analyzed to identify them and separate them from other surrounding objects.
- Monitoring of moving objects. In this case, the objects to be monitored will typically only remain stationary during relatively short periods of time, and will therefore often be present in just a single image. Prior knowledge limiting their possible locations may be available.

An overview of the proposed methods and the applications addressed is given below.

1.4.1 Methodological contributions

In this thesis, three novel methods that can exploit VHR SAR data for the monitoring of man-made objects have been developed.

Method for estimating the dimensions and fill level of oil storage tanks

This thesis presents a new method for the automatic estimation of all the relevant parameters of the cylindrical tanks that are commonly used to store large quantities of petroleum products (e.g., crude oil). Most of these oil storage tanks are located above ground [Pullarcot, 2015] and can easily be observed in satellite images due to their large sizes. Their geometry is well known, as they are built following standards such as the API Standard 650 [American Petroleum Institute, 2013]. Two types of oil storage tanks can be distinguished: those with a fixed or a floating roof. Floating roofs rise and fall with the amount of fluid inside the tanks in order to decrease the vapor space above the liquid level, and are preferred for the storage of highly volatile fluids [Pullarcot, 2015]. For a given storage tank, the proposed method will estimate its maximum capacity and determine whether it has a fixed or a floating roof. For tanks with a floating roof, the amount of oil stored will also be estimated. The proposed method can extract all the relevant information from a single VHR SAR image. However, if a time series is available, all the images can be processed jointly, exploiting the available temporal information to provide more accurate and robust estimates than those obtained from each individual image. Even though it is focused on a specific application, this method illustrates how prior knowledge on an object's geometry and approximate location can be exploited.

Method for the detection of changes associated to man-made objects

In this thesis, a novel unsupervised change detection method for the monitoring of man-made objects is presented. This method will detect changes caused by the appearance, disappearance, or movement of objects inside the imaged scene, as well as changes to static objects. Rather

than looking for changes in SAR amplitude or the loss of coherence, changes are detected by the appearance and disappearance of the strong point scatterers present in man-made objects and often denoted as coherent scatterers (CSs) [Sanjuan-Ferrer et al., 2015]. This enables the detection of changes involving man-made objects while ignoring changes to natural targets such as vegetation. These CSs are detected in each image and compared coherently across an image pair or a time series. An object-based change analysis step is introduced to identify changes significantly larger than individual CSs. When using a time series, the proposed method can categorize the changes according to their temporal behavior. While this method works with an image pair and can detect changes by objects present in a single image, it is especially well suited for the monitoring of objects which are present in multiple images. Therefore, this method benefits from the availability of time series with high temporal resolution, like those acquired by the new SAR missions with large satellite constellations and a high revisit.

Method for the detection and classification of objects

This thesis introduces a new method for object recognition in single VHR SAR images. The proposed approach is based on supervised learning and therefore requires a training dataset with enough samples of the objects to be detected. Because deep learning methods typically require large amounts of training data and these are often not readily available and expensive to generate for VHR SAR data, special focus has been placed in developing a method that can be trained with a limited amount of data without overfitting. Instead of using a typical network architecture for object detection, the task is formulated as a template matching problem, and a fully convolutional Siamese network architecture [Bertinetto et al., 2016] is used. After the training process, a template database is generated by selecting a set of representative training samples of the objects to be detected (e.g., samples for the different imaging geometries). During inference, these templates are then used to search and detect these objects in new images. A significant advantage of the proposed approach based on template matching, is that it could be adapted for one-shot learning: a single sample of a new unknown object could be used as a template to detect more instances of this object, potentially without requiring any additional training.

1.4.2 Applications addressed in this thesis

The three methods proposed in this thesis have been applied to address the three following applications.

Monitoring of oil inventories

The monitoring of the fill level of oil storage tanks serves as an example on the monitoring of well-known, static objects. Because of the relevance that oil production has in the economy, this has recently become a popular commercial application of satellite data [Orbital Insight, nd; Ursa Space, nd]. By applying the method proposed in this thesis to a time series of a refinery with floating roof tanks, its oil production can be effectively monitored. Given that spaceborne SAR missions can regularly acquire images of many different locations across the globe, the proposed method could potentially be applied for the automatic monitoring of oil inventories.

Monitoring of construction activity

As an example on the monitoring of unknown static objects in complex scenes, the unsupervised CD method proposed in this thesis will be applied to monitor construction activity in urban areas. The detection of changes due to the construction of new buildings and infrastructure or renovations to existing ones is of interest for many applications. In addition to these changes, many

other changes (e.g., seasonal vegetation changes, snow, moving cars, etc.) are also continuously occurring across urban areas. Most general CD methods simply result in a binary change map highlighting all the changes. In contrast, the CD method proposed in this thesis can exploit time series to identify specific changes by their characteristic temporal behavior. This method will be used to detect the final change to each newly constructed or renovated buildings and the date when it happened, which should correspond to the time when the construction work was finished.

Monitoring of airport activity

As an example on the monitoring of moving objects, two of the methods developed in this thesis are applied for monitoring airplane traffic at airports. This is relevant for security-related applications and can also provide an insight into economic activity. In this case, prior knowledge delimiting the possible locations of the airplanes (e.g., on aprons, taxiways, runways, etc.) can be easily obtained, as airports tend to be accurately mapped and rarely change. The unsupervised CD method proposed in this thesis can be applied to detect changes due to the movement of man-made objects inside these regions, most of which will correspond to arrival and departure of aircrafts. However, this method is not able to distinguish different types of moving objects (e.g., different types of aircrafts, maintenance trucks, etc.) or to separate closely packed objects (e.g., for counting the number of airplanes). To avoid these limitations, the proposed object detection method is trained to detect and classify the different types of aircrafts present in each image. Nevertheless, both methods are complementary. The CD method can provide valuable information on the arrival and departure times of the detected airplanes, as it can determine whether an airplane parked in the same spot in two images moved or remained stationary. Besides, it can also be applied to detect activity associated to objects for which no training data is available.

1.5 Research questions

By applying the proposed methods to address the aforementioned applications, this thesis also aims to answer the following research questions:

- How robustly and accurately can a method using high resolution SAR images automatically estimate the amount of oil in storage tanks?
- To what extent can an unsupervised method exploiting SAR time series identify the changes corresponding to specific types of objects or events?
- How much training data is required to accurately and robustly identify specific objects in VHR SAR images using a deep convolutional neural network, and how well can the trained network generalize to different locations or imaging geometries?
- How can the temporal information obtained from change detection complement and improve the performance of other methods which typically analyze a single image?

1.6 Structure of thesis

The reminder of this thesis is organized as follows:

Chapter 2 provides an overview of the state of the art for the different types of methods and applications that will be tackled in this thesis.

In Chapter 3, some fundamental characteristics of spaceborne SAR data are briefly described. Subsequently, some of the methods which are typically employed for processing this data are

introduced, as some of them are then applied as intermediate processing steps in the methods proposed in this thesis. Finally, some additional data sources that can be used to gain some prior knowledge about the scene are briefly introduced.

Chapter 4 presents the method to estimate all the relevant parameters of oil storage tanks using either a single VHR SAR image or a time series.

Chapter 5 presents the method for the detection of changes associated to man-made objects in time series of VHR SAR images.

Chapter 6 presents the method for object detection and classification in a single VHR SAR image using deep learning.

Chapter 7 describes the experiments performed to test the methods proposed in this thesis. The experiments for the methods in Chapters 4, 5 and 6 are presented in Sections 7.1, 7.2 and 7.3, respectively. In each section, the data used to evaluate the corresponding method is first introduced, and suitable values for the method's parameters are selected based on this data. Finally, the main experiments are introduced, describing how these methods were applied to solve specific problems involving the monitoring of different types of human activity.

In Chapter 8, the results obtained for the different experiments are shown. Section 8.1 shows the results for the monitoring of oil storage, Section 8.2 for the monitoring of construction activity, and Section 8.3 for the monitoring of airport activity.

In Chapter 9, the outcome of the performed experiments is discussed, and the suitability of the proposed methods for monitoring different types of human activity is evaluated.

Finally, Chapter 10 concludes this thesis, answering the research questions based on the results of the performed experiments, and outlining future research directions to continue this work.

2 State of the art

2.1 Monitoring of oil storage tanks with satellite images

Both SAR and optical satellite images are often employed for the monitoring of oil storage. The appearance of an oil storage tank with a floating roof in both optical and SAR images can be seen in Fig. 2.1. For comparison, a tank with a fixed roof is shown in Fig. 2.2. When using optical images acquired close to nadir angle, such as the one shown in Fig. 2.1a, the position of a tank's floating roof can be estimated from the size of the shadow created by the tank's wall over this roof (since the sun's position at the time of the image acquisition is known). On the other hand, in off-nadir (i.e., oblique) optical images and SAR images, the position of the floating roof can be directly seen, as shown in Fig. 2.1b and Fig. 2.1c, respectively. Specifically, in SAR images the vertical displacement of the floating roof with respect to the bottom of the tank can then be directly measured by exploiting the layover effect.

During the acquisition of the optical images, sunlight illumination and a cloud-free sky are required. Because the satellites acquiring these images are on LEO, each of them can only image a given location on Earth at a few specific instants every several days. If the required sunlight illumination or atmospheric conditions are not given during these time slots, the monitoring of the oil storage tanks is not possible. This severely limits the applicability of methods employing optical imagery, especially for some locations where clouds and smog are often present. In the literature, most of the publications dealing with oil storage tanks in optical remote sensing images focus exclusively on the detection of such tanks [Ok & Baseski, 2015; Zhang et al., 2015; Wang et al., 2016b; Liu et al., 2019; Jing et al., 2019; Sheng et al., 2020; Wu et al., 2022; Xu et al., 2022]. Only two publications have been found which present automatic methods for estimating the volume of storage tanks: Wang et al. [2016a] estimate the amount of oil stored in each tank from the shadow in the floating roof, and Wang et al. [2019] estimate each tank's maximum capacity using the shadow casted outside the storage tank.

(a) (b) (c)

Figure 2.1: Appearance of an oil storage tank with a floating roof in satellite images. a) Nadir optical image, b) oblique optical image, c) SAR image. Both optical images were obtained from Google Earth, and the SAR image with TerraSAR-X.

(a) (b) (c)

Figure 2.2: Appearance of an oil storage tank with a fixed roof in satellite images. a) Nadir optical image, b) oblique optical image, c) SAR image. Both optical images were obtained from Google Earth, and the SAR image with TerraSAR-X.

Unlike optical sensors, SAR sensors can acquire images day-and-night, and those in X-band and lower frequency bands are practically not affected by clouds, fog or smog. Therefore, regular measurements of oil production can be ensured. The achievable accuracy depends on the sensor's spatial resolution, the chosen imaging geometry and the quality of the algorithm used to extract this information. However, this last point represents a challenge: due to radar specific imaging effects such as speckle noise, a robust and automatic extraction of this information from SAR images is far more challenging than in the case of optical images.

In the literature, several methods have been presented for the detection of oil storage tanks in SAR images [Xu et al., 2014; Zhang et al., 2019; Zhang & Liu, 2020], and there are four publications dealing with the estimation of the dimensions and the amount of oil stored in such tanks [Guida et al., 2010; Hammer et al., 2017; Anahara & Shimada, 2018; Back & Jeon, 2020]. Guida et al. [2010] show that it is possible to accurately estimate the height and diameter of cylindrical oil storage tanks with fixed roofs from SAR images, and that this can be achieved by exploiting the imaging geometry as well as its radiometry. They also mention that the same principle could be applied to estimate the fill level of oil storage tanks with a floating roof. In [Hammer et al., 2017], a detailed analysis of the SAR signature of oil storage tanks with both fixed and floating roofs has been performed using SAR simulation, and the authors have additionally shown how all the relevant dimensions of the storage tanks can be estimated by measuring the distance between a few points on their SAR signature and taking into account geometric effects such as layover. The accuracy of these measurements has been verified using a TerraSAR-X image and ground truth data about the dimensions and fill level of the storage tanks, which was provided by the company operating the imaged oil tanks. An important limitation of the approaches presented in these two papers [Guida et al., 2010; Hammer et al., 2017] is that these measurements must be performed by a human operator, which has to manually select some specific image points for each oil storage tank. This greatly limits the applicability of these approaches for regularly monitoring the fill level of a large number of oil storage tanks.

More recently, two publications [Anahara & Shimada, 2018; Back & Jeon, 2020] have introduced methods capable of automatically extracting some information about oil storage from SAR images. The method proposed by Anahara & Shimada [2018] can automatically estimate the vertical displacement over time of the floating roof of oil storage tanks using time series of moderate resolution SAR images. For this, a reference point is automatically selected for each storage tank by analyzing the minimum and average amplitude of the time series, and the location of the amplitude peak corresponding to the double reflection at the tank's floating roof is detected for each individual image. The vertical displacement of the floating roof with respect to the reference point can then be obtained from the shift of this amplitude peak along the range axis. Back &

Jeon [2020] present a method capable of estimating the dimensions and the amount of oil stored in tanks with a floating roof using a single high resolution SAR image. For this the image patch containing an oil storage tank is first denoised, and then some specific points of the storage tank are detected using two one dimensional amplitude profiles along the azimuth and range axes. The dimensions of the storage tank are then derived from these points. However, a description of the algorithm applied to automatically detect the points used to estimate the tank radius is not provided. The method was tested using KOMPSAT-5 SAR data, and the obtained results were compared with those estimated from optical images acquired by KOMPSAT-3.

In conclusion, all the methods employing optical images are affected by external factors (e.g., Sun and clouds), and regular observations with optimal accuracy cannot be guaranteed. These limitations are imposed by inherent characteristics of optical sensors that cannot be easily changed. On the other hand, when using SAR images for monitoring the fill level of oil tanks, the main limitations are not imposed by the sensor, but rather by the accuracy and robustness of the available algorithms that can be employed for extracting this information. The complexity of automatic information extraction from SAR images can be seen by the fact that from the four methods that have been presented in the literature which deal with the estimation of oil storage, only the methods presented in [Anahara & Shimada, 2018; Back & Jeon, 2020] are capable of extracting some of this information automatically without the need of manual interaction. The method introduced in [Back & Jeon, 2020] will be considered as the state of the art, as it is the most recent publication and also capable of automatically extracting more information on oil storage.

In Chapter 4 of this thesis, a new method for the automatic and precise monitoring of oil storage using high resolution SAR images is presented, which has some significant advantages when compared to the current state of the art:

- It can be applied for both storage tanks with floating and fixed roofs, and can automatically classify tanks according to their roof type. The method described in [Back & Jeon, 2020] only works with tanks with a floating roof.
- It exploits the complete geometry of the storage tanks by taking into account all the points in their semicircular double reflections, rather than just using a few select image points as in [Back & Jeon, 2020]. This increases the robustness and the accuracy that can be achieved.
- If multiple images are available, the proposed method can exploit them jointly to increase the accuracy of each measurement and the overall robustness of the method, rather than simply performing multiple individual measurements.
- It can track the vertical displacement of all the point scatterers in a floating roof with subpixel accuracy, which allows the precise estimation of changes in oil storage over time.

2.2 Change detection with satellite images

Change detection (CD) with Earth observation (EO) images is based on the idea that the differences between two images of the same scene acquired at different times will be either due to changes that occurred in this scene, or due to certain effects like different illumination conditions and/or clouds. If the differences that do not depend on real changes occurred on the ground can be mitigated (e.g., during a pre-processing step), then a simple comparison between the two images will allow the detection and mapping of the occurred changes. The development of novel methodologies for change detection with multitemporal EO data has been a subject great interest in the last years [Bovolo & Bruzzone, 2015], which can be seen by the sharp increase in the

number of journal articles on this topic. These techniques have been applied to a great variety of EO data, including that acquired by spaceborne SAR and optical sensors.

Recent research on change detection with optical satellite images has been focused on the development of supervised methods based on deep learning, often using Siamese networks [Chen et al., 2021a; Fang et al., 2021; Zhu et al., 2023]. These methods must be first trained on a dataset with annotated changes, but can then accurately identify the changes in the imagery that correspond to actual changes on the ground.

In contrast, most of the recent work on SAR change detection has been focused on the development of unsupervised methods. Unsupervised methods are especially interesting, because they do not require a dataset with annotated changes. Such datasets are not abundant and creating them is difficult and time consuming, as in most cases changes will need to be manually annotated, because ground truth information is often either not available, costly, or impossible to collect [Bovolo & Bruzzone, 2015]. SAR sensors are better suited for unsupervised change detection because they can ensure a consistent illumination of the same scene at different times, even during adverse atmospheric conditions (e.g., clouds).

Many of these unsupervised CD methods using SAR images employ traditional approaches based on signal processing and statistical analysis. As previously introduced, coherent (CCD) and incoherent (ICD) methods can be distinguished. For example, Su et al. [2015] presented an ICD method for high resolution images which detects changed pixels by applying a likelihood ratio test after a multitemporal pre-denoising step. This method also attempts to categorize the detected changed pixels into different classes (step, impulse, cycle and complex changes) according to their temporal behavior along the time series. Mendez Dominguez et al. [2018] presented an ICD method that exploits multisquint processing to analyze the spatial response of targets in high resolution images. This allows to reduce false alarms due to image artifacts such as sidelobes and azimuth ambiguities, while preserving most of the detection rate of traditional schemes. Monti-Guarnieri et al. [2018] presented two CCD methods (a space coherent, time incoherent method, and a full space and time coherent method), both based on the generalized likelihood ratio test, and demonstrated them using Sentinel-1 time series.

More recently, some authors have proposed unsupervised CD methods for SAR data that employ supervised learning approaches, but are trained in an unsupervised manner by automatically generating pseudolabels. For example, Gao et al. [2021] presented a Siamese adaptive fusion network to extract high-level semantic features of multitemporal SAR images, and trained it using pseudolabels generated by the method described in [Gao et al., 2016]. Dong et al. [2022] proposed a different approach, integrating unsupervised clustering with a convolutional neural network (CNN) in an unified framework, and optimizing the CNN feature learning and clustering end-to-end without supervision.

A significant disadvantage of most unsupervised CD methods using SAR data is that they cannot easily distinguish different types of changes, such as those caused by man-made objects and seasonal changes like vegetation growth or snow. Such seasonal changes are not relevant for applications focused on man-made objects, but they will be detected anyway by traditional coherent and incoherent change detection methods, as they can induce significant amplitude changes and coherence loss. For the particular case of snow, significant changes can occur across the whole imaged scene even in a short time span. Such seasonal changes can dominate the change detection results, making the resulting change maps of little use [Reigber et al., 2016]. Even modern methods such as deep learning based approaches will be affected by this issue if they are simply trained to detect changes in SAR amplitude, which is the typical approach employed in many papers [Li et al., 2019b,c; Geng et al., 2019].

In addition to the previous limitations, conventional CCD and ICD methods cannot unambiguously distinguish changes due to the appearance and disappearance of man-made objects. For CCD methods, both the appearance and disappearance of a man-made object cause the coherence to drop significantly, and therefore these types of events cannot be distinguished by evaluating the coherence. On the other hand, ICD methods can distinguish changes caused by an increase or decrease in the SAR amplitude. Typically, the appearance of a new man-made object in the scene will cause a strong increase in the SAR amplitude at the corresponding location, whereas its disappearance will lead to a decrease in amplitude. ICD methods often exploit this, assuming that the changes detected due to an increase/decrease in SAR amplitude are caused by the appearance/disappearance of a new object. However, these assumptions are not always valid, as man-made objects will often cast a radar shadow, and therefore the appearance/disappearance of an object can also lead to a strong decrease/increase of the SAR changes associated to buildings.

Some recent publications have presented SAR change detection methods focused on the detection of changes caused by man-made objects and not those associated to natural targets such as vegetation. These methods tackle this issue in different ways, such as by implementing hand-crafted detectors for specific types of man-made objects [Bovolo et al., 2013], by using polarimetry to detect durable and permanent changes in urban areas [Kim et al., 2016], by using Persistent Scatterer Interferometry (PSI) [Ferretti et al., 2001] to detect new buildings in large time series [Yang & Soergel, 2018], or by decomposing the images in a time series into background and strong scatterers components and using the extracted scatterers for change detection [Lobry et al., 2016a,b]. While these approaches partially solve the aforementioned issue, they also have certain limitations. The method proposed by Bovolo et al. [2013] relies on hand-crafted features and needs to be tuned for different objects. The approach introduced by Kim et al. [2016] requires polarimetric SAR data, which is typically not available in most high resolution (i.e., below 1 meter) spaceborne SAR systems. The application of PSI by Yang & Soergel [2018] requires a very long time series (e.g., more than 20 images) and can only detect very slow or permanent changes caused by objects which remain unchanged along many images. The method presented by Lobry et al. [2016b] also requires a time series, and the used decomposition model can detect at most a single change event per each pixel during all the period covered by the time series. Finally, the approach proposed by Lobry et al. [2016a] detects changes by an increase or decrease in the number of strong scatterers inside a given radius, and appears unable to detect changes where the scatterers change but their number remains stable. Besides, the decomposition model used assumes a constant or slowly changing background component, which might not always be the case.

In Chapter 5 of this thesis, a novel change detection approach is presented for the monitoring of man-made objects using pairs or series of high resolution SAR images. Rather than looking for changes in SAR amplitude or the loss of coherence, these changes will be detected by the appearance and disappearance of the strong point scatterers which are present in man-made objects, and often denoted as coherent scatterers (CSs) [Sanjuan-Ferrer et al., 2015]. The CSs will be first detected in each individual image by analyzing their phase stability over different frequency subbands [Schneider & Papathanassiou, 2009; Giacovazzo et al., 2008], and can then be compared coherently. Because for most applications the goal is to detect changes which are significantly larger than individual point scatterers, object-based analysis will be applied to refine the results from the pixelwise change detection using CSs and to extract information about changes on an object level. The proposed approach is completely unsupervised and does not require any training data, and it avoids many of the previously mentioned limitations of other change detection methods:

- By focusing exclusively on these point scatterers, only changes associated to man-made objects will be detected and changes induced by distributed scatterers (such as vegetation or snow) can be ignored.
- It exploits coherent change detection, being able to detect subtle changes, while working well even with large temporal baselines, as these coherent scatterers are not significantly affected by temporal decorrelation.
- Four different types of change can be distinguished: the appearance of an object, its disappearance, the replacement by a different object or an alteration to an existing object, and no change (i.e., an object remains unchanged and static).
- Unlike the previously listed change detection methods also exploiting strong scatterers [Yang & Soergel, 2018; Lobry et al., 2016a,b], it works with as few as two images, and when using a time series of n images it can detect up to n different CSs per pixel.
- When applied to a time series, irrelevant transient changes (where an object is just temporarily affected by an external factor such as snow and does not actually change) can be identified and ignored.
- The object-based analysis step makes it possible to target specific types of changes (e.g., by their size and/or temporal behavior) and segment the objects causing them.

2.3 Object recognition in satellite images

In the last few years, great progress has been made in the field of object detection in satellite imagery. Different types of neural networks (NN), like the very popular convolutional neural networks (CNN) or newer architectures like Transformers [Vaswani et al., 2017], have successfully been applied to optical and SAR data. Optical imagery is more widely used for object detection tasks, with several private companies offering mature commercial solutions. This is in part thanks to the availability of several large and high quality datasets [Li et al., 2020; Long et al., 2021; Ding et al., 2022] that can be used to train very large models [Wang et al., 2022]. In contrast, there is a lack of high quality open datasets with VHR SAR data, likely due to the difficulty of accurately labelling objects in SAR images, and the fact that VHR SAR images are also rarely openly available. Besides, there are also some challenges specific to SAR images which do not apply to nadir optical imagery. These SAR specific challenges will be described in detail later in Chapter 6. Nevertheless, the field of object recognition on SAR images, also known as SAR ATR, is rapidly progressing, and the number of available datasets (and their quality) is growing. For many years, most ATR methods were evaluated using the popular MSTAR dataset [Ross et al., 1998]. Nowadays, the MSTAR dataset is considered too easy for deep neural networks [Zhu et al., 2021], as multiple methods have achieved near perfect accuracy. For example, Chen et al. [2016] used a 5-layer CNN to achieve an accuracy of about 99% in the MSTAR dataset. This has led to the creation of alternative datasets and a focus on different object types, like airplanes or ships. Most of the recent research has been focused on modifying modern network architectures to take into account the specific characteristics of SAR images. Below, a brief overview of some recent SAR ATR methods will be provided. For convenience, these will be grouped according to their intended application (i.e., the type of objects to be detected).

The detection of ships in SAR images has always been a popular application. Traditionally, constant false alarm rate (CFAR) methods have been applied to detect ships offshore (i.e., in open water), often using medium resolution SAR images. Most recent work is mainly focused on the development of deep learning methods to accurately detect and even segment ships in

more complex situations (e.g., ships inshore and densely arranged), often using data with a higher resolution. Several public datasets, like HRSID [Wei et al., 2020], SSDD [Zhang et al., 2021] or SRSDD [Lei et al., 2021] are available for these tasks. HRSID and SSDD include data from different sensors and resolutions and can both be used for ship detection and segmentation. SRSDD only includes Gaofen-3 data with 1 meter resolution and cannot be used for segmentation, but it contains rotated bounding boxes. HRSID and SSDD consider all the ships as a single class, whereas SRSDD distinguishes six different categories of ships (e.g., cargo ships, fishing boats, etc.) and can therefore also be used for ship classification. Several authors have used these datasets to train and evaluate new ship detection methods, and compare them to other state of the art approaches. Chen et al. [2021b] applied network pruning and knowledge distillation to obtain an accurate, fast and lightweight ship detector, which was evaluated using the SSDD and HRSID datasets. Xia et al. [2022] proposed CRTransSar, a network architecture combining a vision transformer and a CNN, and applied it to ship detection using the SSDD dataset, as well as other custom dataset with additional object classes. Wei et al. [2022] introduced the low-level feature guided network (LFG-Net), an extension of the feature pyramid network, and applied it for ship instance segmentation. Zhou et al. [2023] used a pyramid vision transformer network and applied multiscale feature fusion to detect arbitrarily oriented ships using rotated bounding boxes. A more comprehensive review on the recent advances on ship detection, including an overview of the different types of methods and the available datasets, can be found in [Li et al., 2022].

Recently, the detection of aircrafts has become a very popular topic, with many deep learning methods being specifically developed for this task. Unfortunately, very few public datasets are available for this task, and the majority of the articles published on this topic use custom datasets. This makes the development of new methods more time consuming, and also makes it more difficult to compare the performance of different methods. Recently, Zhang et al. [2022] introduced the public SAR aircraft detection dataset (SADD), including TerraSAR-X images acquired with different imaging modes and resolutions ranging from 0.5 to 3 meters. Zhang et al. [2022] also proposed a new network, named scale expansion and feature enhancement pyramid network (SEFEPNet), as a baseline for this benchmark. Chen et al. [2022] proposed a geospatial transformer framework, based on a multiscale geospatial contextual attention network (MGCAN), and applied it to detect aircrafts in a custom dataset with GaoFen-3 images with a spatial resolution of 1 meter. Kang et al. [2022] presented the scattering feature relation network (SFR-Net), which includes a scattering point relation module to leverage the position information of the discrete point scatterers commonly present in airplane SAR signatures, and demonstrated it using GaoFen-3 data. Zhao et al. [2022] presented a modified single shot detector by including an attention module to fuse multiscale features, and using deformable convolutional kernels to take into account the discrete nature of aircraft SAR signatures. This method was demonstrated using a custom dataset with GaoFen-3 and TerraSAR-X images. Most of the published works, including those listed above, only deal with the detection of aircrafts, considering all types of aircrafts as a single class. While less common, some authors have also aimed to classify the detected aircrafts (i.e., to distinguish different airplane models). To tackle multiclass aircraft detection, Bao et al. [2022] proposed a sparse attention-guided fine-grained pyramid module and a simple copy-paste data augmentation strategy to balance the number of samples across the different airplane classes. Sun et al. [2022a] presented a network for few-shot aircraft classification, named the scattering characteristics analysis network (SCAN). This network uses a scattering extraction module to learn the number and distribution of the scattering points for each target type. Additionally, the authors use a metalearning approach that takes into account the orientation angle of the aircrafts. To test this method, the authors also created a new dataset, named SAR aircraft category dataset (SAR-ACD), which is not publicly available. Overall, airplane detection appears to be a more challenging task than ship detection, as airplane SAR signatures often consist of just a

few isolated point scatterers, and even small changes in an airplane's orientation can significantly alter its SAR signature.

The accurate detection of cars has been recently demonstrated using airborne SAR images with 10 cm resolution, acquired by the miniSAR and FARADSAR sensors [Sandia National Laboratories, nd]. Several authors have used this data, manually annotating the cars with bounding boxes, to train and evaluate different methods for vehicle detection. Du et al. [2020] used a saliency-guided single shot multibox detector that integrates saliency information (which can highlight targets of interest while suppressing clutter) into the network. Shi et al. [2021] proposed an unsupervised domain adaptation strategy, which can be used to adapt an already trained network to perform detection on data acquired by a different SAR sensor (e.g., different frequency band) for which no labels are available. Zou et al. [2022] presented SCEDet, a network that takes three subaperture images as the input for feature extraction, and then applies multiscale semantic feature fusion and self-attention to aggregate the global context information before the detection head. All these methods were able to detect most of the cars in these airborne SAR images. While SAR images acquired by current commercial SAR satellites have a coarser resolution, such an application should also be eventually possible with satellite data, as the resolution of spaceborne SAR sensors keeps improving.

In addition to vehicles, deep learning methods have also been successfully applied for the detection and segmentation of buildings on VHR SAR images. Shahzad et al. [2019] combined a fully convolutional NN with a conditional random field represented as a recurrent NN, training the whole network in an end-to-end manner to accurately detect building regions in TerraSAR-X high-resolution Spotlight images. An automatic technique was applied to annotate the training data, combining TomoSAR point clouds with OpenStreetMap (OSM) data. Sun et al. [2022b] proposed a conditional GIS aware network, named CG-Net, that learns multi-level visual features and employs building footprint data to normalize these features, and used it to tackle the problem of individual building segmentation in large-scale urban areas. The authors generated large amounts buildings labels automatically by combining accurate DEM and GIS building footprints. Sun et al. [2022c] also proposed an efficient bounding box regression network that exploits the location relationship between a building's footprint and its bounding box to retrieve building heights from a single SAR image. Lee et al. [2022] introduced a semi-supervised learning framework with semantic equalization learning and applied it for building segmentation, leveraging labeled SAR and optical image pairs and unlabeled SAR images from the SpaceNet-6 dataset.

In Chapter 6 of this thesis, a novel method for object detection and classification in VHR SAR images is introduced. Instead of using one of the typical network architectures for object detection, as done by the majority of the SAR ATR methods published in the literature, the task is formulated as a template matching problem and is solved by applying similarity learning, using a fully convolutional Siamese network architecture [Bertinetto et al., 2016]. By creating a template database with a set of representative training samples of the different objects to be detected, these objects can then be detected in new images during inference. This type of network is typically used for visual tracking, but both the network architecture and the training strategy will be modified here to adapt it for SAR ATR. To the author's best knowledge, this is the first time that this kind of network has been applied for object detection in SAR images. The proposed approach has several advantages:

- Because the network takes image pairs as input, it can be trained with relatively few labelled samples without overfitting, as even small datasets will allow to generate a significant amount of different image pairs.

- It takes into account the strong effect that the imaging geometry has in the appearance of man-made objects in SAR images, by considering instances of the same object imaged from different geometries as different classes. This allows to accurately classify objects and also estimate their orientation.
- It uses a balanced data sampling strategy during training that improves the detection performance for the less common object classes, and also for the less common imaging geometries.
- It could potentially be used to perform few-shot or one-shot object detection by simply adding the new samples to the template database, since these networks are typically used for tracking arbitrary objects in videos given just a single sample.

3 Fundamentals

3.1 Basics of spaceborne SAR data

Different product types and processing levels

Spaceborne SAR data is usually available in different processing levels. The lowest processing level, typically denoted as level 0, corresponds to the raw data acquired by SAR sensors. Unlike for optical sensors, for SAR sensors the raw data does not give any useful information on the imaged scene until signal processing is applied to form a SAR image [Moreira et al., 2013], a process which is known as SAR processing. For modern spaceborne SAR missions, SAR processing is a complex operation which is typically performed on the ground segment after the raw data is downlinked. Therefore, the raw data is very rarely processed by end users. The most common output of SAR processing is a single-look complex (SLC) image, which is typically considered a level 1 image product. SLC images consist of the real and imaginary parts of the focused complex SAR data (from which the amplitude and phase can be computed), given in a slant-range and azimuth coordinate system. The slant-range axis is aligned with the direction of the radar's line of sight, and the azimuth axis with the direction of the satellite's movement. SLC images have the full spatial resolution obtained after processing all the available bandwidth, and will often have different resolutions (and pixel spacings) in range and azimuth. These image products are meant for further processing and not for visualization, and are used by most scientific users.

Most end users use image products with higher processing levels, containing only amplitude information, which are generated from the raw data or the SLC image. To generate these image products, multilooking is applied to reduce the speckle noise and improve the radiometric resolution, at the cost of a worse spatial resolution. The resolution will typically be equal in range and azimuth, resulting in a square pixel size. This kind of image products are delivered either in a ground-range projection (i.e., range and azimuth coordinate system projected into an Earth ellipsoid model), or a map projection (by applying geocoding, possibly with a DEM to correct distortions due to varying terrain height). These radiometric and geometric corrections make it easier for end users to work with the data.

In this thesis SLC images will be used, as they contain the most information and enable analysis which are not possible with higher level products (e.g., coherent processing of time series). A few basic methods for processing SLC SAR images (e.g., despeckling, geocoding, etc.) will be briefly described in Section 3.2.

Different imaging geometries

Due to the side-looking imaging geometry, spaceborne SAR sensors can only observe a given location using a few different acquisition geometries. Unlike airborne sensors, for which the flight direction can be usually arbitrarily chosen, a satellite's orbit cannot be easily changed, and will determine how many and which imaging geometries are possible for a given location. Satellites in LEO will revisit each location with two different ground tracks, corresponding to an ascending

orbit (i.e., satellite traveling from the south towards the north pole) and a descending orbit (i.e., traveling from the north towards the south pole). The use of different incidence angles will often enable the imaging of a given location from several ascending and/or descending orbits. Using steeper incidence angles will allow to image a location from orbits closer to it, whereas shallower incidence angles will allow to image it from further away. SAR sensors will be able to perform acquisitions with incidence angles within a certain range, which will vary between different sensors. Most sensors should at least be able to operate with incidence angles within the range of 25° to 45°, with some sensors supporting even wider ranges (e.g., TerraSAR-X can operate with incidence angles between 15° and 60°). Additionally, many SAR satellites will also be able to acquire images in left- and right-looking configurations, rotating to have the antenna point in the appropriate direction, which further increases the imaging possibilities. Overall, all the possible imaging geometries for a given location are given by the combination of ascending and descending orbits, different incidence angles and left- and right-looking configurations.

While the use of an ascending or descending orbit and a left- or right-looking geometry will simply determine the direction from which a given scene is imaged, the incidence angle will have additional implications on the acquired SAR images. Different incidence angles will lead to a change in the amount of backscattered energy, altering the appearance and overall characteristics of the SAR images. Normally, shallower (i.e., higher) incidence angles result in a lower backscatter. The change in backscatter for different incidence angles will vary for different targets. Geometric effects such as shadow and layover also depend on the incidence angle, and play a significant role on SAR images containing tall man-made objects (e.g., high buildings in urban areas) and/or large terrain variations. Also, while the slant-range resolution only depends on the available bandwidth, the ground-range resolution will also depend on the used incidence angle, with steeper (i.e., lower) incidence angles leading to a worse ground-range resolution.

SAR time series

The ability to acquire time series with many SAR images of a given location is one of the most powerful capabilities of SAR satellites. Typically, SAR time series are acquired using a repeat-pass orbit (i.e., with a nearly identical ground track), which allows to acquire images with the same imaging geometry and at regular intervals. Time series with repeat-pass acquisitions can be coherently processed, and can therefore be used for interferometry, coherent change detection, etc. The repeat-pass interval of a certain SAR mission will depend on its orbit and the number of satellites. Typically, SAR missions in LEO have an orbital repeat cycle in the order of 10 to 20 days (e.g., 11 days for TerraSAR-X, 12 days for Sentinel-1, 22 days for Iceye, etc.). By using two satellites, the Sentinel-1 mission can achieve a repeat-pass interval of 6 days, and the fast growing Iceye constellation is expected to reach a repeat-pass interval of one day in the near future. For some SAR missions (e.g., Capella) repeat-pass acquisitions are not be possible, as their orbits do not result on a repeated ground track. In this case, time series can still be formed by combining images acquired from different orbits and with different imaging geometries. While coherent processing is not possible in this case, these time series can still be exploited. Depending on how different the imaging geometries are, these time series will be suited for different applications: images with relatively similar imaging geometries will be better suited for incoherent change detection, whereas larger differences can be beneficial for radargrammetry. Even for sensors with a repeat-pass orbit, the combination of images acquired from different orbits can still be of interest, as it will allow to reduce the time between two consecutive image acquisitions of the same location.

Visualization of SAR images

To properly visualize a VHR SAR image, some of its specific characteristics need to be taken into account. The first thing to consider is the very large dynamic range of these images (e.g., up to 90 dB for TerraSAR-X high resolution Spotlight images) [Zhu et al., 2021]. Also, the distribution of amplitude values is extremely asymmetric: typically, the majority of the pixels will contain distributed scatterers and have values in the low amplitude range, with all the remaining amplitude values being only taken by the pixels containing strong point scatterers [Zhu et al., 2021]. Attempting to visualize the complete dynamic range of the SAR amplitude in a linear scale will result in an image which is mostly black and contains a few bright spots where the strong point scatterers are located. A better mapping of such a large dynamic range to a grayscale image can be achieved using a logarithmic scale, expressing the SAR amplitude in decibel (dB). Moreover, this effectively converts the speckle, which appears as multiplicative noise, into an additive noise component. When showing a SAR image in dB scale, it is still recommended to limit its dynamic range, as this results in a grayscale image with a higher contrast with minimal information loss. Because most modern SAR sensors are well calibrated and the amplitude values have a physical meaning, the same fixed dynamic range can typically be used for all the images acquired by a certain sensor and imaging mode. Unless otherwise stated, the SAR images shown this thesis (all TerraSAR-X images acquired using the Staring Spotlight imaging mode [Mittermayer et al., 2014]) will all be shown in dB scale, with all values below -30 dB appearing black (i.e., taking a grayscale value of 0), and those above 20 dB white (i.e., taking a grayscale value of 255).

In addition to the dynamic range, geometric effects such as layover and foreshortening that are inherent to the SAR imaging principle also need to be considered, as these can make image interpretation more difficult. Often, in SLC SAR images, the image's x-axis corresponds to range and the y-axis to azimuth. Besides, SLC images can also have very different resolutions (and therefore pixel spacings) along the range and azimuth axes. In this representation, elevated objects such as buildings fall towards the left or right side of the image (depending on the direction of the range axis) and their dimensions appear distorted. To achieve a more intuitive representation, a SAR image can be resized to achieve a square pixel spacing in slant-range, and rotated so that the range axis corresponds to the image's y-axis (with near-range towards the top). This transformation makes the layover effect easier to interpret, as the resulting SAR image is more similar to an oblique optical image. An example of this transformation, shown for a TerraSAR-X Staring Spotlight acquisition over an urban area with tall buildings, can be seen in Fig. 3.1. This transformations will be applied to visualize all the SAR images shown in this thesis, but the actual processing will be always performed using the original SLC image rasters unless otherwise specified.

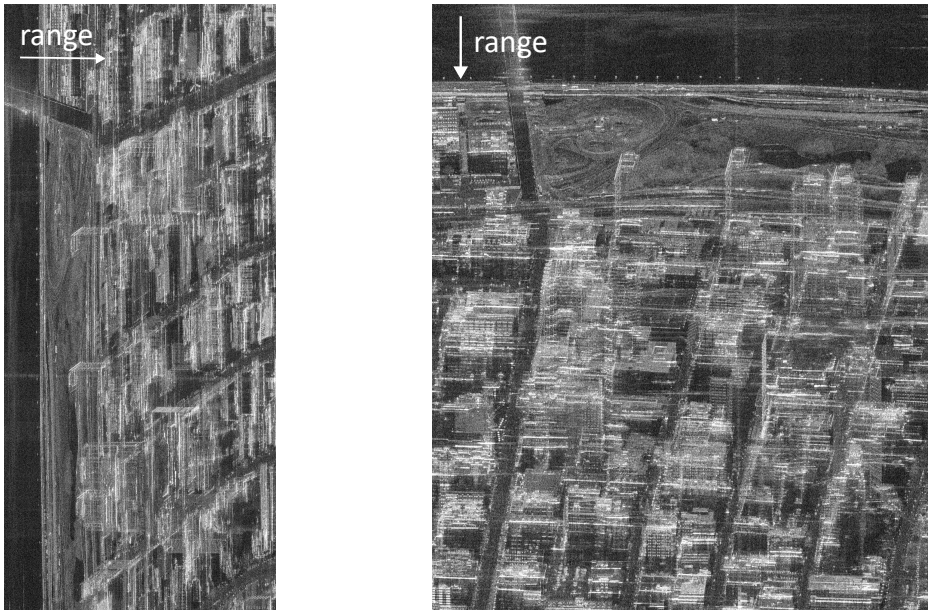
3.2 Processing of SAR images

This Section will provide a brief overview on some of the operations which are typically applied for the processing and analysis of VHR SAR images and time series, and which will also be used in the methods proposed in this thesis. While these are well known by the SAR community, they will be briefly explained here for the reader's convenience.

3.2.1 Single image processing

Sublooking and subaperture processing

In order to form SAR images with a high spatial resolution, a large bandwidth and a long synthetic aperture are required. The acquired raw data is then focused by the SAR processor, resulting in

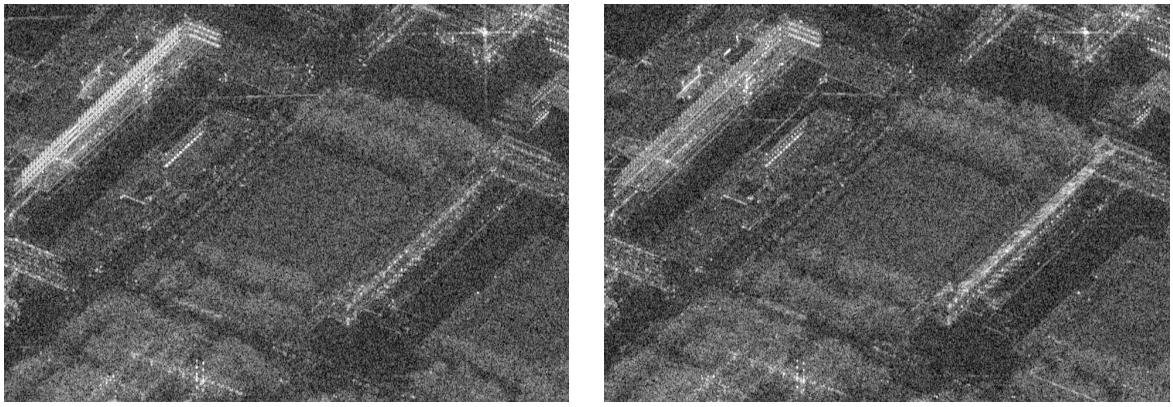


(a) (b)

Figure 3.1: Geometric transformation applied to a SAR image for easier visual interpretation of the layover effect. a) Image patch from an SLC TerraSAR-X Staring Spotlight image, with range in the x-axis and different pixel spacings in slant-range and azimuth. b) Transformed image, with equal pixel spacing in both axes and the layover effect occurring towards the top, allowing an easier image interpretation.

an SLC image. Alternatively, multiple independent SAR images with reduced resolution could also be generated by separately processing the raw data for different frequency subbands and/or subapertures. These images are often referred to as sublooks, and can be used for speckle reduction, the analysis of the scattering behavior for different squint angles, the detection of point scatterers, etc. While ideally these sublooks are generated directly from the raw data, this will not always be possible, as the raw data might not be available (e.g., due to data policies). Besides, it is also impractical, as processing the raw data of modern SAR satellite missions is typically a complex and time consuming operation. However, similar results can be obtained from the full resolution SLC image. A description of the process of computing sublooks from an SLC image can be found in [Sanjuan-Ferrer, 2013]. Nevertheless, it will also be briefly described below for convenience.

Initially, depending on the imaging mode used by the SAR sensor, a deramping (i.e., demodulation) operation might need to be applied to the SAR data to compensate the time-varying Doppler centroid. A brief description of the deramping process for data acquired using the Spotlight imaging mode can be found in [Eineder et al., 2009]. After this, a fast Fourier transform (FFT) should be applied to compute the spectrum of the deramped SLC image. The window function (e.g., Hamming) applied to the spectrum during SAR processing for sidelobe reduction should be then removed (i.e., by multiplying by its inverse). The information on the used window function is typically available on the image metadata, but could alternatively also be obtained from the data. At this point, sublooks can be easily generated by using a band-pass filter to select different portions of the spectrum. Optionally, a window function can also be applied here for sidelobe reduction. In any case, when applying the band-pass filtering, a scaling factor should be applied to compensate the associated change in the energy of the spectrum. Finally, an inverse FFT needs to be applied to compute the sublook image from the filtered spectrum.



(a) (b)

Figure 3.2: Example of sublook images computed using two different subapertures. A change in backscattering for the two squint angles can be seen for the buildings on the left and right sides of these images.

Figure 3.2 illustrates some of the effects that can be observed when analyzing different sublooks. Figures 3.2a and 3.2b show two sublooks obtained from a TerraSAR-X Staring Spotlight image, using two non-overlapping subapertures. Each of these sublook images has only half of the Doppler bandwidth, and therefore an azimuth resolution of 50 cm instead of the 25 cm of the original image. The two subapertures correspond to two independent images of the same scene acquired with two different squint angles and at slightly different times. Therefore, a comparison of different subaperture images like these two can provide additional information on objects whose backscattering changes for different squint angles, as well as on objects moving at the time of the image acquisition. Both effects can be seen in this example. The façades of the buildings on the left (Munich's University of Television and Film) and right (the "Alte Pinakothek" museum) sides of these images exhibit a clearly different backscattering for the two sublooks. The SAR signatures of several objects moving during the image acquisition, which cannot be properly focused during SAR processing, appear in this example as horizontal lines which move between the two sublooks (e.g., those at the upper left corner).

Speckle filtering

Speckle noise makes the analysis and interpretation of SAR images more difficult, and negatively affects the performance of methods for information retrieval. Because of this, speckle filtering, also referred to as despeckling or speckle reduction, represents an important processing step for many applications and is the subject of active research. The most basic approach for speckle reduction is the so-called multilooking, which involves the computation and subsequent non-coherent averaging of multiple sublook images. Alternatively, a similar result can be achieved in the spatial or image domain by applying a boxcar filter to average the amplitude of multiple image pixels. In both cases, the number of averaged samples (either sublooks or pixels) is referred to as the number of looks. A higher number of looks will enable a better speckle reduction, at the cost of a reduced spatial resolution.

More advanced methods exploiting signal and image processing techniques can achieve a significant reduction of speckle noise with no resolution loss. A comprehensive review of such methods, including an overview over many different approaches, can be found in [Argenti et al., 2013]. Besides despeckling, image and signal processing methods can also be applied for achieving a better sidelobe reduction [Stankwitz et al., 1995; Abergel et al., 2018]. In the last years, deep

(a) (b) (c)

Figure 3.3: Example results for different despeckling methods. a) Original SAR image (for reference). b) Multilooking by incoherently averaging multiple overlapping sublooks. c) Advanced filter for speckle reduction without resolution loss.

learning methods, such as the one proposed by Dalsasso et al. [2022], have also been successfully applied for SAR despeckling, achieving state of the art performance.

The effect of speckle reduction is illustrated in Fig. 3.3, which shows the results obtained when applying two different despeckling methods to an SLC TerraSAR-X Staring Spotlight image (with a resolution of approx. 60 cm in slant-range and 25 cm in azimuth). These results are visualized for a small image patch, which shows Munich's Bavaria statue and the adjacent building. The original image patch, affected by speckle noise, can be seen in Fig. 3.3a. Figure 3.3b shows the results obtained by applying multilooking, incoherently averaging many overlapping sublooks to achieve a resolution of 1.2 meters in slant-range and azimuth. Here, it can be clearly seen that while multilooking is able to significantly reduce the speckle noise, the coarser spatial resolution leads to a loss of detail. Finally, the results obtained by applying a more advanced speckle filter can be seen in Fig. 3.3c. This filter is able to achieve a good speckle reduction without any loss in resolution, making the resulting image more suitable for further analysis. This particular speckle filter has not been published and will not be described here, as it is outside of the scope of this thesis. However, state of the art despeckling methods like MERLIN [Dalsasso et al., 2022] should provide even better results.

Detection of coherent scatterers

Strong point scatterers, also known as coherent scatterers (CSs), can be detected in a high resolution single-look complex (SLC) SAR image by applying spectral diversity techniques [Sanjuan-Ferrer et al., 2015]. Multiple methods have been presented in the literature for the detection of CSs [Schneider et al., 2006; Schneider & Papathanassiou, 2009; Giacovazzo et al., 2008; Sanjuan-Ferrer et al., 2015]. All of these methods are based on the same principle: they exploit the high bandwidth available in high resolution images by computing multiple sublook images with lower spatial resolution, and then detect CSs by identifying those targets which remain stable across these sublooks. The main difference between these methods is the criterion used to determine which targets remain stable across the multiple sublooks. A comparison of the performance of the different methods can be found in [Sanjuan-Ferrer et al., 2015]. In this thesis, the phase variance approach (PVA) [Schneider & Papathanassiou, 2009] will be used for CS detection, due to its simplicity and relatively good performance. This method will be briefly described below.

The basic idea behind the CS detection using PVA is that if a given image pixel contains a CS its phase will vary linearly with sublook frequency, whereas if not it will vary randomly.

Therefore, multiple sublooks with a reduced bandwidth and different central frequencies should be computed, so that the phase trend can be analyzed for each pixel. In this thesis, the sublooks will be computed along range (i.e., corresponding to different frequency subbands), as is typically done for CS detection [Schneider et al., 2006; Schneider & Papathanassiou, 2009; Giacovazzo et al., 2008; Sanjuan-Ferrer et al., 2015]. The same detection method could also be applied to azimuth sublooks [Sanjuan-Ferrer et al., 2015], but the performance is expected to be worse in that case, due to the fact that many point scatterers have a nonconstant azimuth angular scattering pattern [Schneider et al., 2006].

For a given pixel, its phase at each sublook can be denoted as ϕ_i, with $i = 1, ..., n$, where n is the number of sublooks. To avoid issues when analyzing the phase trend due to phase jumps of 2π around $\pm\pi$, phase unwrapping should be performed by adding or subtracting multiples of 2π as required so that $|\phi_{i+1} - \phi_i| \leq \pi$. If a pixel's phase varies linearly with frequency, the change in phase between every two consecutive sublooks should be nearly constant and therefore have a low variance. This variance, which will be denoted here as σ_ϕ^2, can be computed as follows:

$$\mu_\phi = \frac{1}{n-1} \sum_{i=1}^{n-1} \phi_{i+1} - \phi_i \tag{3.1}$$

$$\sigma_\phi^2 = \frac{1}{n-1} \sum_{i=1}^{n-1} (\phi_{i+1} - \phi_i - \mu_\phi)^2 \tag{3.2}$$

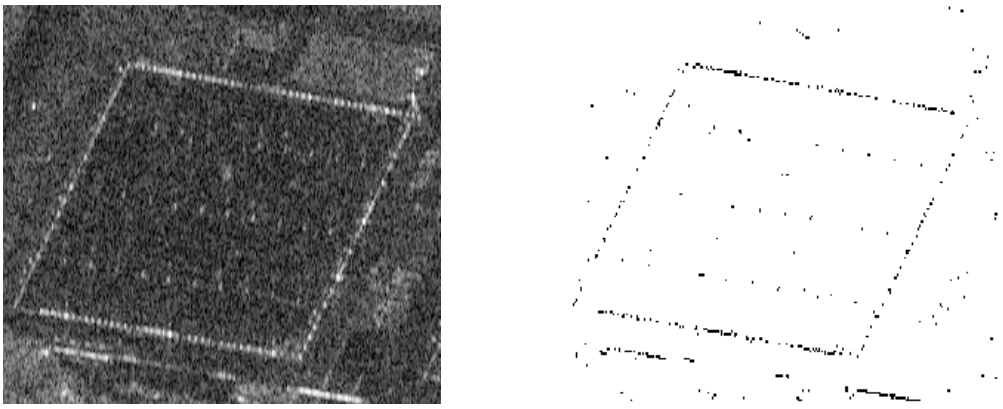
CSs can be detected by computing σ_ϕ^2 for each pixel and applying a threshold, which can be denoted as T: pixels with $\sigma_\phi < T$ will be considered to contain a CS. The value of T trades off the number of false positives and false negatives.

For the pixels with a linear phase, the slope of this phase ramp will be directly related to the distance along the range axis between the center of that pixel and the actual location of the corresponding CS [Sanjuan-Ferrer et al., 2015]. If μ_ϕ is used as a estimation of the slope of this linear phase ramp, this distance, which can be denoted as d_r, can be expressed as:

$$d_r = \frac{c}{4\pi f_s} \mu_\phi \tag{3.3}$$

where c is the speed of light and f_s is the difference between the central frequencies of two consecutive sublooks. This relation can be exploited to determine the location of CSs along the range axis with a higher accuracy than that given by the range resolution of the SAR sensor. Additionally, it can also be used to mitigate the resolution loss introduced by sublook computation, which will cause strong point scatterers to spread to neighboring pixels. In such cases, those neighboring pixels might also have a linear phase trend even if they do not contain a CS, and would be falsely identified as CSs when just applying a threshold to σ_ϕ^2. However, because their phase ramps will have a steeper slope, these can be filtered out by also thresholding μ_ϕ. The corresponding limits for μ_ϕ can be analytically derived by enforcing that a CS must be inside the resolution cell of a pixel (i.e., less than half a pixel distance between the CS and the pixel center): $|d_r| \leq 0.5\ \delta_y \Rightarrow |\mu_\phi| \leq \frac{2\pi\delta_y f_s}{c}$.

Figure 3.4 shows an example of the results that can be obtained when applying the described method to TerraSAR-X Staring Spotlight data. For this example, the following parameters were used: a threshold $T = 0.5$, and $n = 40$ sublooks covering the whole available bandwidth with a 75% spectral overlap between consecutive sublooks. For the used TerraSAR-X imagery with 300 MHz total bandwidth, this resulted on a sublook bandwidth of 27.90 MHz and a separation of $f_s = 6.97$ MHz between sublooks. The selected parameters result in a good detection performance,



(a) (b)

Figure 3.4: Example results for the detection of coherent scatterers. a) SAR amplitude. b) Detected CSs.

as most of the point scatterers visible in the SAR image are correctly detected, and almost no CSs are detected in the clutter areas. Besides, the image with the detected CSs has virtually the same resolution as the input SLC image.

Geocoding and orthorectification

For each pixel in a SAR image, the geographic coordinates describing the corresponding position on the Earth's surface can be precisely determined by applying geocoding. In a similar way, given the geographic coordinates of a certain location, inverse geocoding can be applied to find the corresponding image pixel. The geometry of a SAR image acquisition is accurately described in an Earth-centered, Earth-fixed (ECEF) frame by the so-called range-Doppler equations [Curlander, 1982], which relate the two-dimensional SAR image coordinates range and azimuth to a point in 3-D space. The range equation uses the slant-range measured by the SAR sensor to relate the target's unknown position to the sensor's position. The Doppler equation relates the Doppler shift induced by the movement of the sensor with the target's unknown position and the sensor's position and velocity. In the case of a single SAR image (i.e., no interferometry or radargrammetry), a third equation, based on the assumption that the imaged scatterers lie on the Earth surface, is required to solve the system of equations and compute the ECEF coordinates for a given pixel. This equation is typically based on a simple Earth model like the World Geodetic System 1984 (WGS84) ellipsoid. Optionally, the ellipsoid equation can be modified to take into account the terrain height at a given location. Solving these equations does not require any knowledge of the satellite's attitude or the antenna pointing. For any given pixel, the Doppler shift, slant-range, the sensor position and its velocity can be obtained from the image metadata, and their values are typically very accurate for modern SAR missions. This system of equations will have two possible solutions, corresponding to left- and right-looking geometries, and care should be taken to ensure that the correct solution is selected. A more detailed description of the range-Doppler equations and the different errors sources affecting geocoding accuracy can be found in [Gisinger, 2019].

The range-Doppler equations are typically solved iteratively, which can make geocoding time consuming. Faster and easier geocoding can be achieved by using rational polynomial coefficients (RPC), a generic sensor model that can be used to substitute the range-Doppler equations with negligible loss of accuracy [Zhang et al., 2011]. To compute the RPCs for a SAR image, the range-Doppler equations must be first solved for a sparse grid of pixels distributed across the image, to obtain their geographic coordinates for different elevation values. The unknown parameters of

the RPCs can then be determined by solving a system of equations using the points in this grid. Different RPCs need to be computed to perform direct and inverse geocoding, but these can be computed using the same grid. Using the resulting RPCs, the geocoding process is reduced to the evaluation of four polynomials (typically of second or third degree) and a few trivial arithmetic operations.

Even when using RPCs, direct geocoding with a DEM will remain an iterative process, as the terrain height for a given pixel is a priori unknown. After performing geocoding with a certain height value, the actual terrain height for the resulting coordinates can be obtained from a DEM. If the height value used for geocoding and the DEM height differ, the solution will be invalid, and geocoding will need to be performed again with the newly obtained height value. This process needs to be repeated until convergence, which will typically occur after a few iterations.

Orthorectification can be applied to transform SAR images into a map projection, enabling the direct measurement of positions on the ground and correcting terrain-related geometric distortions. The use of a sensor independent geometry also facilitates the combination with other types of data. To generate an orthorectified SAR image, an image grid with a similar pixel spacing and covering the extension of the imaged scene needs to be generated in the desired map projection. For each of the pixels in this grid, the terrain height can be obtained from a DEM, and inverse geocoding can then be applied to find the corresponding pixel in the SAR image. The SAR amplitude values for the pixels in this image grid can then be interpolated from the original SAR image.

3.2.2 Time series processing

Co-registration

Co-registration is a process that aims to achieve a precise alignment between two or more SAR images of the same scene to facilitate a joint analysis. Typically, a reference image (also denoted as master image) is selected, and the remaining secondary images (also called slave images) are resampled to match the reference image. This results in an image stack where each pixel represents the same location in all images. This process can be applied to SAR images acquired by different sensors, with different imaging geometries and/or at different times. However, because the focus of this thesis is the processing of SAR time series, in this work co-registration will be applied to series of SAR images acquired at different times, by the same satellite mission (i.e., nearly identical sensors) and with a nearly identical imaging geometry (i.e., repeat-pass orbit). To enable a coherent analysis of such time series, a very accurate co-registration (i.e., subpixel accuracy) is required. In most cases, very accurate co-registration can be achieved simply by applying forward and backward geocoding, as nowadays precise orbit information and an accurate DEM are typically available. However, small biases might remain in both range and azimuth (e.g., due to timing errors), which need to be estimated from the data [Sansosti et al., 2006]. For this, tie points across the imaged scene can be generated by applying geocoding, and image patches at these tie points can be cross-correlated to get multiple estimates of these biases, which can then be combined using least-squares or a similar method. After this estimation, the reference images can be accurately resampled by applying geocoding and accounting for the estimated azimuth and range biases.

SAR images acquired with certain imaging modes will have a time-varying Doppler spectrum. This needs to be taken into account when co-registering SLC images by modulating the kernel used to interpolate the complex data [Eineder et al., 2009]. Alternatively, deramping can be applied prior to the interpolation step, which can then be performed using a regular kernel, optionally followed by a reramping operation.

Coherence calculation

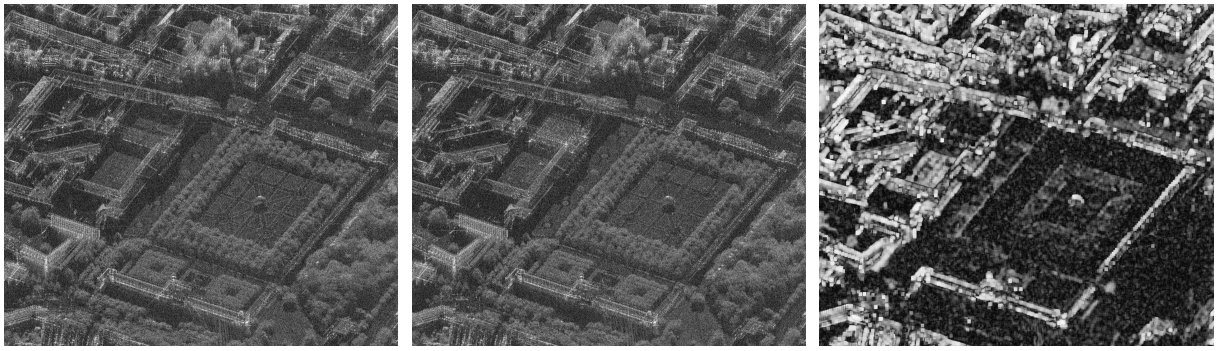
The complex correlation between two interferometric SAR images is known as interferometric coherence or simply coherence [Bamler & Hartl, 1998; Rosen et al., 2000]. Although this is actually a complex variable, the term coherence is often used to refer to its magnitude. In this thesis, the term coherence will always refer to the magnitude of this complex correlation. The coherence plays an important role on the analysis of interferometric time series (i.e., with repeat-pass acquisitions), as it is a measure for local interferogram quality [Bamler & Hartl, 1998]. Besides, low coherence is also a good indicator for changes in the imaged scene [Preiss et al., 2006]. Given two co-registered SLC images, the coherence, denoted as γ, can be computed as follows:

$$\gamma = \left| \frac{\langle x_1 x_2^* \rangle}{\sqrt{\langle x_1 x_1^* \rangle \langle x_2 x_2^* \rangle}} \right| \tag{3.4}$$

where x_1 and x_2 denote the two complex SAR images, and $\langle \ldots \rangle$ indicates the spatial averaging of multiple samples (i.e., multilooking) which is required to get a good coherence estimate. This estimate is biased, as will tend to overestimate low coherence values, especially when using a low number of samples to estimate the coherence. However, it will become asymptotically unbiased for large numbers of samples. The topic of coherence estimation is discussed in detail in [Touzi et al., 1999]. Typically, the spatial averaging is performed using a boxcar filter, and the resolution of the resulting coherence map will depend on the kernel size (which controls the number of averaged samples). However, more advanced methods have also been presented, which aim to obtain more accurate coherence estimates and avoid the resolution loss induced by the spatial averaging. Many of these methods are based on advanced signal and image processing techniques (e.g., NL-SAR by Deledalle et al. [2015]), but more recently deep learning methods (e.g., Φ-Net by Sica et al. [2021]) have also been successfully applied for this task, achieving state of the art performance.

When using repeat-pass images, the time between the two image acquisitions is typically denoted as temporal baseline. The temporal baseline plays an important role, as the coherence of certain targets (especially distributed scatterers like vegetation) will drop for longer temporal baselines. This phenomenon is known as temporal decorrelation, and will be more significant at higher frequencies like X-band [Parizzi et al., 2009]. Small changes in the imaging geometry will also lead to a coherence loss, an effect known as geometric (or baseline) decorrelation. The change in imaging geometry will induce a spectral shift [Gatelli et al., 1994], and its effects on the coherence can be mitigated in post-processing by filtering both complex images to a common bandwidth. A much more detailed description of the interferometric coherence, the different decorrelation sources, and SAR interferometry in general can be found in [Bamler & Hartl, 1998; Rosen et al., 2000].

Figure 3.5 shows some example results for the coherence estimation and also illustrates the effect of temporal decorrelation. Figures 3.5a and 3.5b show two TerraSAR-X Staring Spotlight images of the city of Munich, acquired with a temporal baseline of 11 days. The coherence for this image pair can be seen in Fig. 3.5c. In this example, the coherence was estimated using a boxcar filter with a kernel size of 9×23 (in range and azimuth, respectively), resulting a resolution of approximately 5.2 meters in both axes. Different decorrelation rates can be clearly observed for the different surfaces. The buildings and other static man-made objects show a high coherence, as these are not significantly affected by temporal decorrelation. Areas with vegetation exhibit a very low coherence, showing that, at X-band, after 11 days most natural land cover classes are almost completely decorrelated.



(a) (b) (c)

Figure 3.5: Example results for the coherence estimation. a-b) Two SAR images acquired with a temporal baseline of 11 days. c) Interferometric coherence.

4 Monitoring of oil storage tanks using coherent scatterers

This Chapter introduces a novel method for the automatic estimation of all the relevant parameters of oil storage tanks. For a given storage tank, the proposed method will estimate its maximum capacity and determine whether it has a fixed or a floating roof. For tanks with a floating roof, the amount of oil stored will also be estimated. This method can be applied for monitoring changes in oil storage, by observing many storage tanks with a floating roof using series of SAR images acquired at different times. Part of the material in this Chapter has been published in [Villamil Lopez & Stilla, 2021].

This Chapter is organized as follows: first, in Section 4.1 the SAR signatures of oil storage tanks with floating and fixed roofs will be briefly analyzed, showing how their relevant parameters can be derived from the semicircular double reflections in them. Additionally, some constraints which simplify the detection of these semicircular double reflections will be established. Then, a method that exploits this to automatically extract all the relevant information for a given storage tank will then be presented. This method will be described for two different cases: one in which a single SAR image is available, which will be described in Section 4.2, and another one in which a time series of SAR images is available, presented in Section 4.3.

4.1 SAR signature of oil storage tanks

Oil storage tanks have a simple and well-known geometry and they exhibit a very characteristic SAR signature, which has been analyzed in detail via SAR simulation by Hammer et al. [2017] for both tanks with fixed and floating roofs. Multiple semicircular double reflections are present on the SAR signatures of both types of tanks. Knowledge of these semicircles is sufficient to estimate all the relevant parameters of an oil storage tank: its precise location and maximum capacity, and also its current fill level if the tank has a floating roof. Therefore, if these semicircles can be detected in a SAR image, they can be exploited to extract precise information about oil storage.

This section will analyze the geometry of both types of storage tanks, as well their temporal behavior and backscattering properties, and establish some constraints which can be enforced to simplify the detection of these semicircles. As the monitoring of oil storage tanks with a floating roof is the most interesting use case, their SAR signature will be analyzed first and in more detail. Then, the signature of an oil tank with a fixed roof will be briefly analyzed by comparing it to the signature of a tank with a floating roof.

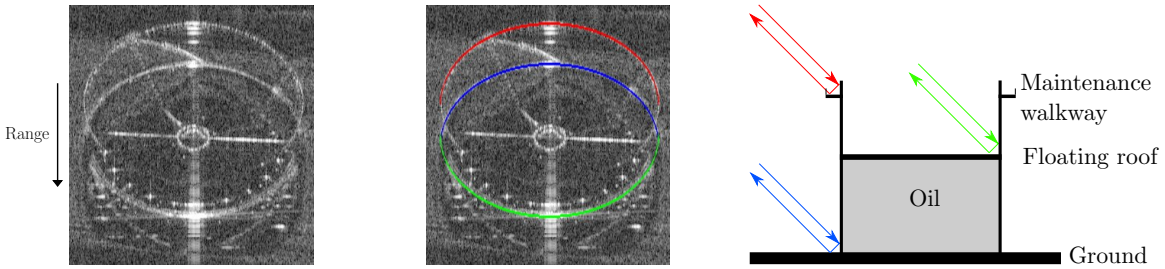


Figure 4.1: SAR signature of oil storage tanks with a floating roof and its three characteristic semicircular double reflections. a) Amplitude SAR image, b) amplitude SAR image with highlighted semicircular double reflections, c) illustrative drawing of the geometry.

4.1.1 Oil storage tanks with a floating roof

A SAR image showing an oil storage tank with a floating roof can be seen in Fig. 4.1a. This SAR image has been acquired by TerraSAR-X with an incidence angle of 48.1°, a resolution of 58 cm in slant-range and 23 cm in azimuth. As it can be seen in Fig. 4.1a, the SAR signature of an oil storage tank with a floating roof contains three bright semicircles, which appear as semi-ellipses in the image due to the chosen pixel spacing (i.e., corresponding to a slant-range projection). These are highlighted in Fig. 4.1b, and correspond to the three double reflections illustrated in Fig. 4.1c:

- One occurring between the outer wall of the cylindrical tank and the ground (shown in blue in Fig. 4.1b and 4.1c).
- Another between the outer wall of the tank and a maintenance walkway located near the top (shown in red in Fig. 4.1b and 4.1c).
- The last one between the inner wall of the tank and the upper surface of the floating roof (shown in green in Fig. 4.1b and 4.1c).

In this work, these reflections will be referred to as semicircles, as this is their true shape, even though they typically appear as semi-ellipses in SAR images. The semiaxes of the ellipses in the SAR image can be easily calculated from the radius of the oil tank r_t, the mean incidence angle θ with which the corresponding SAR image was acquired, and the pixel spacing along the x axis (δ_x) and y axis (δ_y). Without loss of generality, it will be assumed that the x and y image axes correspond to the azimuth and range axes of the SAR image, respectively. The semiaxes of this ellipse r_{tx} and r_{ty}, expressed in image pixels, can be computed with the following equations:

$$r_{tx} = \frac{r_t}{\delta_x} \tag{4.1}$$

$$r_{ty} = \frac{r_t}{\delta_y} \sin\theta \tag{4.2}$$

When attempting to detect these semicircles in a SAR image, the following constraints, which are imposed by the geometry of the problem, can be enforced:

- The three semicircles have the same radius, as it can be assumed that the thickness of the tank's wall is negligible.

- Their centers lie along the same vertical axis, located at different heights. Therefore, the three semicircles will have the same center along the azimuth axis, but they will exhibit a translation in range due to the layover effect.
- Due to the side-looking imaging geometry of SAR sensors, only the side of the tank's outer wall which is closer to the SAR sensor can be imaged. Therefore, the two semicircles corresponding to the double reflections with the outer tank wall will appear towards near-range. These are highlighted in red and blue color in Fig. 4.1.
- Regarding the tank's inner wall, only the side which is farther away from the SAR sensor can be seen in the SAR image. Therefore, the double reflection corresponding to the inner wall of the tank (highlighted in green in Fig. 4.1) will appear towards far-range.

The aforementioned displacement of the semicircles centers along the range axis due to layover can be exploited to compute the height of the cylindrical tank h_t and the vertical position of the floating roof h_r by using the following equations:

$$h_t = \frac{l_t \delta_y}{\cos\theta} = \frac{(y_b - y_t)\delta_y}{\cos\theta} \tag{4.3}$$

$$h_r = \frac{l_r \delta_y}{\cos\theta} = \frac{(y_b - y_r)\delta_y}{\cos\theta} \tag{4.4}$$

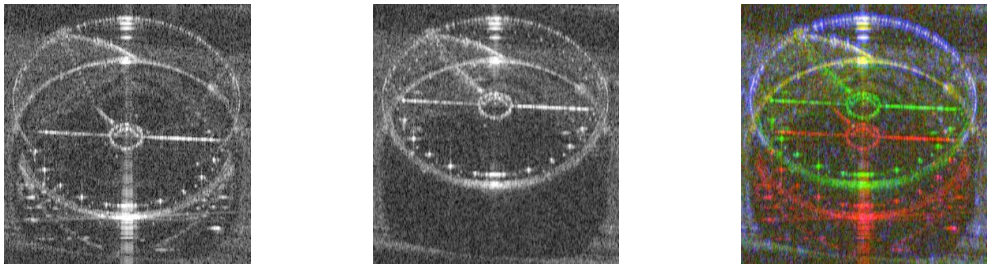
where l_t and l_r represent the layover (in pixels) due to the height of the cylindrical tank and the vertical position of the floating roof, respectively. These can in turn be expressed as a function of y_b, y_t and y_r, which are the pixel coordinates along the y axis (range) of the centers of the semicircular double reflections at the tank bottom, top, and floating roof, respectively. Both equations assume that the range axis of the SAR image starts at near-range (such as in all the images shown in this thesis), and will result in negative height values if the SAR image has an inverted range axis (i.e., starting at far-range), as in that case the layover effect will occur towards the opposite direction.

It is important to note that the height of the tank h_t might be slightly underestimated, as it is obtained from the double reflection occurring at a maintenance walkaway, which is slightly below the top of the tank. This error in the height of the cylindrical tank will only affect the estimation of its maximum capacity and not of the amount of oil stored inside it, which is the most interesting parameter. Additionally, this slight underestimation of a tank's height should be easy to compensate: if ground truth data is available for a few storage tanks, it should be possible to derive a offset and/or a scale factor to calibrate the estimated heights, as most oil storage tanks have standard sizes.

If a time series of SAR images acquired at different times is available, additional constraints associated with the temporal behavior can be imposed:

- The double reflections which correspond to the oil tank's outer-wall (red and blue semicircles in Fig. 4.1b) will remain constant over time, as the tank's outer structure does not change.
- On the other hand, when observed at different times the double reflection associated to the oil tank's inner-wall (green semicircle in Fig. 4.1b) can move along the range axis, as the floating roof will rise or sink with changes in the amount of oil stored.

This is illustrated in Fig. 4.2 for a pair of SAR images of the same oil storage tank acquired 11 days apart. The two individual amplitude images are shown in Fig. 4.2a and 4.2b. A



(a) (b) (c)

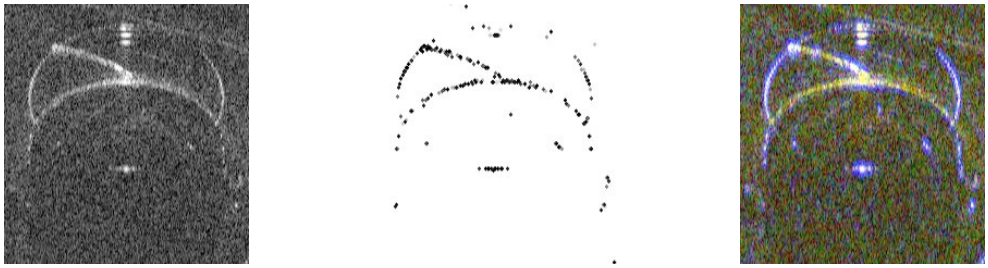
Figure 4.2: SAR images of an oil storage tank with a floating roof at two different dates. a) First image, b) second image, c) multitemporal color composite image with the two amplitude images in the red and green channels, and coherence in blue.

(a) (b)

Figure 4.3: Coherent scatterers detected for an oil storage tank with a floating roof. a) SAR amplitude image, b) detected coherent scatterers.

multitemporal color composite image with both amplitude images in the red and green channels, and the interferometric coherence in the blue channel is shown in Fig. 4.2c. Here, it can be clearly seen how the double reflection corresponding to the floating roof moves between both image acquisitions, which appear in green and red colors in Fig. 4.2c due to the strong changes in SAR amplitude and the low coherence. On the other hand, the two double reflections corresponding to the outer tank structure remain unchanged and appear therefore in blue/white colors, due to the high interferometric coherence and lack of change in amplitude. The outer tank structure is expected to remain unchanged and have high coherence even during long time periods. However, there is no guarantee that the floating roof will move, especially for short time periods, and could also appear unchanged. Therefore, both possibilities (the floating roof moving or remaining static) must be taken into account when exploiting the temporal information in a SAR time series for the monitoring of oil storage.

Finally, in addition to these geometric and temporal constraints, the backscattering properties can also be exploited for simplifying the detection of these semicircles. Many of the pixels in these semicircular double reflections appear in high resolution SAR images as CSs, which can be detected as described in Section 3.2.1. By detecting these CSs in the SAR images, the detection of these semicircles can be greatly simplified. The reason for this is that the majority of the surroundings of the oil storage tank will be eliminated, leaving only a few relevant point scatterers which can be used to fit the semicircles. An example of the CSs detected for an oil storage tank with a floating roof is shown in Fig. 4.3.



(a) (b) (c)

Figure 4.4: SAR signature of an oil storage tank with a fixed roof. a) SAR amplitude image, b) detected coherent scatterers, c) multitemporal color composite image showing temporal change across an image pair acquired 11 days apart.

4.1.2 Oil storage tanks with a fixed roof

A SAR image of an oil storage tank with a fixed roof can be seen in Fig. 4.4a. This image and the one of a tank with a floating roof shown in Fig. 4.1a are part of the same TerraSAR-X image, and have therefore the same resolution and imaging geometry. By comparing both images it can be seen that the SAR signatures of both types of storage tanks are relatively similar, with the exception of the features corresponding to the floating roof. The same two semicircular double reflections at the bottom and top of the cylindrical tank are present, whereas the one corresponding to the floating roof is missing. This implies that the geometrical constraints established in the previous point regarding those two semicircles also apply to storage tanks with a fixed roof, and their height can also be computed using equation 4.3.

When observing a fixed roof tank at different times there will be no significant change, as no floating roof is present, and the two semicircular double reflections corresponding to the outer tank structure will remain unchanged. This can be seen in Fig. 4.4c, which is a multitemporal color composite image highlighting the changes between a pair of images acquired 11 days apart, such as the one shown previously in Fig. 4.2c for a tank with a floating roof. The only changes which can be seen are small amplitude changes in the clutter areas due to speckle noise. The pixels corresponding to the oil storage tank have a high interferometric coherence and no significant amplitude change, and appear therefore in blue and white colors.

The backscattering properties are also similar to those of a tank with a floating roof, and the pixels on the semicircular double reflections are also CSs. As in the previous case, the detection of these point scatterers allows to eliminate the tank's surroundings and can simplify the fitting of these semicircles. The detected CSs for this storage tank with a fixed roof are shown in Fig. 4.4b. Here, it can be seen that fewer CSs are detected at the double semicircular reflection at the top of the tank than in the case of a tank with a floating roof. However, there are still enough point scatterers to fit the corresponding semicircle and estimate the tank height using equation 4.3.

All these similarities can be exploited to develop a method capable of extracting all the relevant parameters of both types of tanks. Additionally, the main difference, which is the lack of the third semicircular double reflection corresponding to the floating roof, can be exploited to easily distinguish both types of tanks. Such a method, capable of classifying both types of tanks and extracting all their relevant parameters out of a high resolution SAR image, will be introduced in Section 4.2. Then, in Section 4.3, this method will be extended to the case in which a SAR time series is available, exploiting the temporal information to achieve more accurate and robust estimates than those possible using each individual image.

4.2 Oil storage estimation from one SAR image

In the previous section it has been established that if the semicircular double reflections of an oil storage tank can be extracted from a SAR image, all its relevant parameters can be derived from them. In this section, a method which exploits the detection of coherent scatterers (CSs) and the geometric constraints described in the previous section to automatically detect these semicircular double reflections will be presented. After these double reflections have been detected for a given storage tank, a simple machine learning classifier can be used to determine whether this tank has a floating or a fixed roof. Given one high resolution SAR image of a oil refinery, the proposed method can be applied to automatically estimate the maximum capacity of each storage tank, as well as the amount of oil stored in the tanks with a floating roof.

4.2.1 Approximate location of the oil storage tanks

Before attempting to detect the semicircular double reflections of a given storage tank, its approximate location in the SAR image should be known. As the goal of the proposed method is not the detection of oil storage tanks, but rather the automatic and precise estimation of their dimensions and of the amount of oil stored in the tanks with a floating roof, it will be assumed that the approximate locations of the oil tanks are known. Here, a brief overview of the different methods and data sources which can be used for obtaining the locations of the storage tanks will be provided.

One possibility would be to obtain these locations from complementary data sources. OpenStreetMap (OSM) data is open and freely available and contains the locations of many of such oil tanks. For the oil refineries which are not present in OSM, the locations of their storage tanks could also be detected in the SAR images using any of the approaches that have already been presented in the literature for the detection of cylindrical storage tanks [Xu et al., 2014; Zhang et al., 2019; Zhang & Liu, 2020]. Alternatively, the oil tanks could be detected also in an optical image if one is available, using one of the methods described in [Ok & Baseski, 2015; Zhang et al., 2015; Liu et al., 2019; Jing et al., 2019]. If enough optical imagery is available, a CNN could also be trained to detect the oil tanks, as described in [Guo et al., 2018; Wu et al., 2022; Xu et al., 2022]. For this, a large training dataset can be semi-automatically generated by using OSM data to obtain the location of many oil storage tanks, generating in this way the labels for the optical or SAR imagery. In addition to the approximate location of the storage tanks, most of the listed methods also provide a bounding box around each tank, which can be used to estimate an approximate value for its radius. While the proposed method does not require an initial estimate for the radius of a given tank, it can be used if available to reduce the computing time and slightly increase the robustness of the method, as it will be shown later.

Independently on the method used to obtain these locations, this step only needs to be performed once, as the locations of such oil tanks change very rarely, and the precise location and other parameters will then be extracted from the SAR image using the method described below.

4.2.2 Estimation of the precise location and size of oil tanks

Once the approximate locations of the storage tanks in the SAR image are known, the CSs in the SAR image can be detected as described in Section 3.2.1. After this, the semicircular double reflections of each tank should be detected to obtain information about oil storage. For a given storage tank, the goal is to determine the following parameters: its radius r_t, height h_t, and vertical position of the floating roof h_r (if there is one). Additionally, the exact pixel position for the center of the tank bottom $p_b = (x_b, y_b)$ will also be determined, from which the corresponding geographic coordinates can easily be computed by applying geocoding. Out of these parameters,

the maximum capacity of the storage tank V_{max} (given in cubic meters) can be easily computed using equation 4.5. If the tank has a floating roof, the volume of oil V_{oil} stored inside it at the time the SAR image was acquired can also be computed in the same way:

$$V_{max} = \pi r_t^2 h_t \tag{4.5}$$

$$V_{oil} = \pi r_t^2 h_r \tag{4.6}$$

Both equations assume that the volume available for oil storage begins at the height of the lower semicircular double reflection, which is expected to occur at ground height. This assumption should be valid for aboveground storage tanks (the most common type of tanks) [Pullarcot, 2015], as the thickness of the tank's bottom plate is expected to be negligible [American Petroleum Institute, 2013]. However, a bias could possibly be introduced in some cases by a foundation of unknown height below the tank: Hammer et al. [2017] observed a bias in the tank fillings estimated from a SAR image, which they attributed to a foundation height of approximately 1 m.

For the tank radius r_t and height h_t minimum and maximum possible values can be specified, taking into account the possible sizes for these storage tanks. These will be denoted as r_t^{min}, r_t^{max}, h_t^{min} and h_t^{max}. These limits for the tank height h_t can be translated into the corresponding limits for the layover l_t (in pixels) using equation 4.3. If the method previously used for obtaining the approximate location of the storage tanks also provides an approximate value for their radius $\hat{r}_t$, it can be used for setting tighter limits for r_t, specified individually for each tank. Later, the results and runtimes will be compared for the cases in which the generic limits for r_t are used, with those obtained when an approximate radius $\hat{r}_t$ is available for each tank.

Besides these size limits, the only actual input required is the approximate pixel position for the center of the tank bottom $\hat{p}_b$, which should have been obtained using any of the methods discussed in the previous subsection. Additionally, an uncertainty value u giving the maximum expected error (in meters) of this approximate position should be specified. This uncertainty u, can be scaled into pixels for both the x axis (u_x) and y axis (u_y) using the respective pixel spacings and incidence angle, in the same way the radius r_t was scaled in equations 4.1 and 4.2.

Using the approximate location of the storage tank $\hat{p}_b$ and its associated uncertainty u, as well as the established limits for the tank radius r_t and layover l_t, a small image patch containing the oil storage tank can be obtained by setting the following boundaries for x and y:

$$\hat{x}_b - u_x - r_{tx}^{max} \leq x \leq \hat{x}_b + u_x + r_{tx}^{max} \tag{4.7}$$

$$\hat{y}_b - u_y - r_{ty}^{max} - l_t^{max} \leq y \leq \hat{y}_b + u_y + r_{ty}^{max} \tag{4.8}$$

The CSs detected inside this image patch can then be used to fit the semicircles corresponding to the double reflections of the storage tank.

Initially, the two semicircles at the bottom and at the top of the outer tank structure (highlighted in blue and red colors in Fig. 4.1) should be detected, as these are present for both the storage tanks with a fixed and a floating roof. Both semicircles appear towards near-range (i.e., towards the top of the y axis in the shown examples), and are defined by the parameters x_b, y_b, r_t and l_t: both have a radius r_t (given in meters), the lower semicircle has its center at pixel (x_b, y_b), and the upper one has its center at pixel $(x_b, y_b - l_t)$. Their detection, and therefore the estimation of these four parameters, can be formulated as an optimization problem: the goal is to find the values for these parameters which maximize the number of CSs that fit the corresponding semicircles. While a brute-force search could be applied to test all the possible parameter values inside their respective intervals, a more efficient implementation is possible by using a Hough

transform. Below, a brief explanation will be provided on how a Hough transform can be used for the detection of these two semicircles on the SAR image.

The Hough transform will compute a three dimensional accumulator array $h(x, y, r)$, whose values will give the number of CSs fitting the semicircle of radius r centered at (x, y). All the elements of this accumulator array must be initially set to zero, and will be updated iteratively. For a given radius r_0 and a CS located at pixel p_i, $h(x, y, r_0)$ should be incremented by 1 for all the pixels (x, y) which could be the center of a semicircle of radius r_0 passing through p_i. Here, the fact that semicircles appear as semi-ellipses in the SAR image must be accounted for. This process should be repeated for all the detected CSs, and for multiple values of the radius r covering the interval between r_t^{min} and r_t^{max}. For sampling the interval for the radius r, it is recommended to select a step δ_r which ensures that the corresponding change in image pixels is equal to or smaller than one pixel for both the x and y axes:

$$\delta_r = \min(\delta_x, \delta_y / \sin\theta) \tag{4.9}$$

The accumulator array $h(x, y, r)$ will tend to have lower values for smaller semicircles, even if they represent a good fit. This can be compensated by computing a new accumulator array $h'(x, y, r)$ which accounts for this:

$$h'(x, y, r) = h(x, y, r)/\sqrt{r} \tag{4.10}$$

Even though the number of CSs along each semicircle grows linearly with the radius r, the normalization was performed by dividing by a factor of $\sqrt{r}$ instead, as this has empirically shown to provide the best results for both small and large semicircles. From $h'(x, y, r)$, a new accumulator array to detect two semicircles with a displacement of l pixels along the y axis can be easily defined:

$$h_2(x, y, r, l) = h'(x, y, r) + h'(x, y - l, r) \tag{4.11}$$

Finally, the values of x_b, y_b, r_t and l_t can be obtained from $h_2(x, y, r, l)$:

$$(x_b, y_b, r_t, l_t) = \underset{\substack{x,y,r,l \\ x\in[\hat{x}_b-u_x,\hat{x}_b+u_x] \\ y\in[\hat{y}_b-u_y,\hat{y}_b+u_y] \\ r\in[r_t^{min},r_t^{max}] \\ l\in[l_t^{min},l_t^{max}]}}{\arg\max}\ h_2(x, y, r, l) \tag{4.12}$$

At this point, the center pixel of the tank bottom and its radius are known, and the tank height h_t can then be easily computed from the obtained layover l_t using equation 4.3. This applies for both tanks with a floating or a fixed roof. However, for tanks with a floating roof, its vertical position h_r is still unknown.

4.2.3 Estimation of the floating roof position

The vertical position of the floating roof h_r can be easily computed from the corresponding layover l_r, which can be obtained by detecting the semicircle due to the double reflection of the floating roof, shown in green in Fig. 4.1. This semicircle also has a radius r_t, its center is at $(x_b, y_b - l_r)$ and, in contrast to the two previous semicircles, it appears towards far-range (i.e., towards the bottom of the y axis in the shown examples). As only one parameter (l_r) needs to be determined, there is no need to apply a Hough transform to detect this semicircle. Instead, cross-correlation with a semicircular binary mask can be performed, displacing it along the range axis (y) over all the possible center locations. For this, the minimum and maximum possible values of l_r need to

be established, which can be easily obtained from the corresponding limits for the floating roof height h_r. If a tank is completely empty, the floating roof will be at the bottom, and therefore $h_r^{min} = 0$. On the other hand, if it is full, the floating roof will be at the top, and $h_r^{max} \simeq h_t$. As previously mentioned, the tank height h_t is obtained from a double reflection at a maintenance walkaway near the top of the tank, and is therefore slightly underestimated. Additionally, if the radius r_t obtained from the Hough transform is slightly larger than the actual radius of the storage tank, a small error will be induced in the estimated center y_b, which in turn will cause to slightly overestimate l_r. To account for this, it is recommended to set a upper limit for h_r which is larger than h_t.

The cross-correlation between these two binary images will give the number of CSs which match each semicircle inside the defined height interval. The best fitting semicircle, and therefore the layover l_r (in pixels) due to the vertical position of the floating roof, can be obtained by finding the maximum of this cross-correlation. By denoting the function with the cross-correlation values for each layover displacement l as $f(l)$, this can be formulated as:

$$l_r = \underset{l \in [0, l_r^{max}]}{\arg\max} f(l) \tag{4.13}$$

where l_r^{max} is the maximum possible layover value, computed from h_r^{max} using equation 4.4.

If a storage tank is very tall and has a comparatively small radius, the floating roof might not be visible if it is located near the tank's bottom and the SAR image was acquired with a flat incidence angle, as it will be hidden by the radar shadow caused by the cylindrical outer wall. In order to be able to properly detect the semicircular reflection of the floating roof when this is located at the bottom, the following constraint can be established: $\tan\theta \leq {}^{1.5r}/_{h_t}$. As long as this condition is satisfied, an arc of around 80° of this semicircle should be visible. In real scenarios this is not expected to significantly limit the applicability of the method, as this condition should be easily fulfilled for most tanks with conventional sizes: any incidence angle smaller than 50° would work for all the storage tanks in the dataset used in this thesis (which is described later in Chapter 7). From this analysis it follows that using SAR images acquired with a steep incidence angle is advantageous for this method, as this allows to see better into the inside of the tank, and additionally it also allows to derive the heights from the layover with a higher accuracy.

A priori it is often unknown whether a tank has a fixed or a floating roof. In such cases the estimation of h_r can be performed as described, even if this only makes sense for tanks with a floating roof. If a given tank has a fixed roof, the cross-correlation value obtained when estimating h_r will be very low, as the double reflection of the floating roof is not present. This information can then be used to determine whether a tank has a fixed or a floating roof, as it will be described below.

4.2.4 Classification of storage tank type

As described in Section 4.1.2, the main difference between the SAR signature of a storage tank with a floating roof and one with a fixed roof is that in the latter there is no semicircular double reflection towards far-range (as it is caused by the floating roof). This will imply that when attempting to estimate the vertical position of the floating roof by detecting its double reflection (as done in the previous step), the number of matched CSs will vary significantly depending on whether the storage tank has a fixed or a floating roof: for a fixed roof this number will be very low, and for a floating roof it will be much higher. This number is given by $f(l_r)$, as defined in Section 4.2.3. A further difference between both types of storage tanks is that for a tank with a fixed roof, less CSs are detected at the double reflection at its top than at the reflection at the bottom, as shown in Fig. 4.4b. On the other hand, for a tank with a floating roof the number of CSs

detected on these two double reflections is similar, as it can be seen in Fig. 4.3b. The number of CSs at the bottom and top double reflections are given by $h(x_b, y_b, r_t)$ and $h(x_b, y_b - l_t, r_t)$ respectively, as defined in Section 4.2.2.

Taking all this into account, the number of detected CSs at each of these three double reflections will be used as the input features for a machine learning classifier such as a support vector machine (SVM) or a random forest. For each storage tank, the corresponding three dimensional feature vector v can be defined as:

$$v = (f(l_r), h(x_b, y_b, r_t), h(x_b, y_b - l_t, r_t)) \tag{4.14}$$

Because only three simple but informative features are used, it will be possible to train an accurate classifier with little training data. Additionally, as oil storage tanks have standard shapes and sizes, a classifier trained with storage tanks located at a given refinery should also be applicable to storage tanks at different locations.

4.3 Oil storage estimation from SAR time series

In the previous section, a method to estimate all the relevant parameters of oil storage tanks out of a single high resolution SAR image has been introduced. An interesting application of such a method is the monitoring of changes in oil storage, by observing many storage tanks with a floating roof using series of SAR images acquired at different times. In such a scenario in which a SAR time series is available, the presented method can be improved by enforcing the temporal constraints established in Section 4.1. This section will show how the additional observations can be exploited to make the method more robust, enable more accurate estimates of several relevant parameters, and improve the classification between tanks with a fixed roof and those with a floating roof. To achieve this, a simple change detection method will be applied to separate the CSs which remain static (e.g., those corresponding to the outer structure of a storage tank) from those that move over time (e.g., those corresponding to a floating roof). Then, the previously introduced method will be extended to take this temporal information into account.

4.3.1 Identification of the static and moving parts of the oil tanks

If two or more repeat-pass acquisitions are available, coherent change detection can be applied to determine which CSs remain static and which move during the corresponding time period. Even though computing the interferometric coherence involves spatial averaging using a small window (which may contain clutter in addition to the CS), strong point scatterers tend to have high coherence values in high resolution SAR images and they do not exhibit significant temporal decorrelation [Ferretti et al., 2001; Eineder et al., 2009]. Therefore, as long as a CS remains static the corresponding pixel should have a high coherence, whereas even a small displacement (i.e., of less than one pixel) will cause a significant drop in the coherence. This principle will be exploited here to separate the static and moving parts of the oil tanks, but later in Chapter 5, it will be extended and developed into a general change detection method.

Given a time series of n SLC SAR images, these should be first co-registered with subpixel accuracy, and the CSs should then be detected for each image in the stack as described in Section 3.2. This will result in a stack of n binary images C_i, with $i = 1, ..., n$, which will have a value of 1 for the pixels containing a CS and 0 for the rest. To detect which of these CSs have moved during the period covered by the time series, a threshold can be applied to the interferometric coherence. If the coherence drops below this threshold for a pixel containing a CS it will imply that this CS has moved, whereas if it is always above this threshold it will mean that it has remained static.

To implement this, the interferometric coherence must be computed for the $n-1$ consecutive image pairs, obtaining a stack with the respective coherence images Γ_j, with $j = 1, ..., n-1$. Then the minimum coherence value can be computed for each pixel, obtaining another image Γ_{min}. In the same way, the maximum of the binary images C_i should also be computed pixelwise, obtaining another binary image C_{max} showing all the pixels for which at least one CS has been detected in the time series. A binary image S with the CSs which remain static during the time series can then be computed by applying the following equation:

$$S(x,y) = \begin{cases} 1 & \text{if } \Gamma_{min}(x,y) > \gamma_t \text{ and } C_{max}(x,y) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

where γ_t is the chosen threshold for the interferometric coherence. A further implication of equation 4.15 is that if a CS always has a coherence above γ_t, it will be considered that it has remained static and it is therefore present in all the images, even if this CS was detected in just a single image of the series. The reasoning behind this is that a high coherence value should only be possible if there is no change, whereas false negatives during the CS detection are much more likely to occur.

Finally, by comparing the static CSs in S with those detected in each individual image C_i, a set of new images M_i showing the CSs that moved can be obtained:

$$\underset{i=1,...,n}{M_i(x,y)} = \begin{cases} 1 & \text{if } C_i(x,y) = 1 \text{ and } S(x,y) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

4.3.2 Estimation of the precise location and size of oil tanks

Once the described change detection method has been applied to identify which CSs remained static and which moved, this temporal information can be exploited when estimating the relevant parameters of oil storage tanks. As in the previous scenario in which a single SAR image was available, initially the size and location of each storage tank should be determined. For this, the approximate locations of the storage tanks in the SAR images should be first obtained as described in Section 4.2.1. Then, a Hough transform can be applied as described in Section 4.2.2 to estimate the parameters r_t (tank radius), h_t (its height), and p_b (the pixel position for the center of the tank bottom). In this case, only the CSs which remain static (i.e., those in the binary image S) should be taken into account when performing this Hough transform. For tanks with a floating roof which moved, this allows to separate the outer tank structure from the floating roof, which in turn makes the detection of the two semicircles faster and more robust, as most of the CSs which do not correspond to these semicircles can be filtered out.

For a given storage tank, once its precise location and size have been determined, what remains is to determine whether this tank has a fixed or a floating roof and, for tanks with a floating roof, the roof position at the acquisition time of each image must be estimated. Again, it will be initially assumed that all the storage tanks have floating roofs and their positions will be estimated, even if this does not make sense for tanks with a fixed roof. The results of this step will be used later to determine whether a storage tank has a floating or a fixed roof. Rather than applying the same approach as in the single image case to estimate the position of the floating roof, this can now be decomposed in two separate steps: the estimation of the vertical displacements of the floating roof between each successive image pair, and the estimation of its initial height.

4.3.3 Estimation of the vertical displacements of the floating roof

When the amount of oil stored in a tank with a floating roof changes, the floating roof will move up or down accordingly. Using a pair of SAR images acquired at different times, the corresponding

vertical displacement of a given floating roof can be directly estimated, without the need to estimate its vertical position in each image.

The vertical displacement of a floating roof between the acquisition times of any two images i and j in the time series will be denoted as d_{ij}, and can be defined as:

$$d_{ij} = l_{r_j} - l_{r_i} \tag{4.17}$$

where l_{r_j} and l_{r_i} denote the layover due to the vertical position of the floating roof at the images j and i in the series. Here the value of d_{ij} is given in pixels, but it can be easily converted into meters in the same way the layover l_r is scaled into the height h_r, using the relation defined in equation 4.4.

If a floating roof moves at some point during the period covered by the time series, its vertical displacement d_{ij} can be estimated using the binary images M_i and M_j, which contain the CSs which moved during the time series and are present at images i and j, respectively. Similar to the previously estimated parameters, the estimation of d_{ij} can also be seen as an optimization problem: the vertical displacement for which more CSs match between the two image patches of M_i and M_j will most likely correspond to the correct solution. This process will match all the features in the floating roof (not only the semicircular double reflection), and can be easily implemented by performing a cross-correlation of these two binary images for different displacements along the range axis (y). For this, a small image patch containing just the corresponding storage tank without any surroundings should be used. The corresponding image region can be obtained by modifying equations 4.7 and 4.8 using the tank location, radius and height values estimated in the previous step, as well as an uncertainty value $u = 0$. Combining equation 4.17 with the limits for l_r previously established in 4.2, it follows that the value of d_{ij} must be within the following range: $-l_r^{max} \leq d_{ij} \leq l_r^{max}$. The maximum cross-correlation value obtained when estimating d_{ij} can be denoted as w_{ij}, and its value will be used at a later step.

If desired, the roof displacements can also be estimated with subpixel accuracy. As described in Section 3.2.1, the location of each detected CS can be determined along the range axis with subpixel accuracy. This can be exploited to compute a subpixel shift, which can be added to the shift obtained from the cross-correlation (which has an accuracy of one pixel). During the cross-correlation step, w_{ij} CSs have been matched between the two images i and j for a displacement of d_{ij} pixels along range. The distance between each of these CSs and the corresponding pixel center can be determined separately for each image by applying equation 3.3. For each one of the matched CSs, the difference between the values obtained from the two images will give an estimate of its subpixel displacement. An accurate estimate of the subpixel displacement for the floating roof can be then obtained by averaging all the estimated subpixel displacements for the w_{ij} matched CSs.

For a time series with n images, the vertical displacement of a given floating roof between each of the $n-1$ consecutive image pairs will provide all the required information on the changes in the amount of oil stored inside the tank. However, the described method can be used to estimate the roof's vertical displacement between each of the $\binom{n}{2} = (n^2-n)/2$ unique image pairs in the series. The additional estimated values will provide some redundant information, as the displacements from multiple image pairs are related (e.g., the displacement between the first and third images must be the sum of the displacements between first and second images, and second and third images). This relation can be formulated in a general way as:

$$d_{ij} = d_{ik} + d_{kj} \tag{4.18}$$

For convenience, it will be assumed that the displacements are computed for the image pairs with $j > i$ (the remaining image pairs with $j < i$ do not provide any additional information).

This redundant information can be exploited to obtain a more robust and accurate estimate of the changes in oil storage, as the displacement values estimated from each individual image pair using cross-correlation may contain small errors, and in some rare cases even a few of these estimates may be significantly off.

These improved estimates of the vertical displacement of the floating roof between each of the $n-1$ consecutive image pairs will be denoted as d'_k, with $k = 1, ..., n-1$. Their values can be obtained by solving an overdetermined linear equation system with $(n^2-n)/2$ equations (one for each estimated d_{ij}) and $n-1$ unknowns (the values of d'_k to be estimated). To formulate this linear system, each of the estimated d_{ij} must be expressed as a linear combination of the unknowns, which can be done using the following equation:

$$d_{ij} = \sum_{k=i}^{j-1} d'_k \tag{4.19}$$

Such a overdetermined linear system can be solved by applying a method such as least squares. In this work, the Huber regressor will be used, which is a robust regression method and less sensitive to outliers than least squares. When performing this regression, the maximum cross-correlation values w_{ij} obtained when estimating each d_{ij} term can be used as weights, as typically those estimates with lower cross-correlation values will be less accurate.

Using this approach, the vertical displacements of a floating roof (and therefore the changes in oil storage) can be estimated with very high accuracy. The larger the number of SAR images in the series, the more robust and accurate that these estimates will become. It is important to note that the estimated values will only be valid if the floating roof moved during the period covered by the time series. However, this is not an issue, as the case in which the floating roof does not move will also be accounted for next, when estimating its initial position.

4.3.4 Estimation of the initial position of the floating roof

Finally, in order to have an absolute measurement of the amount of oil stored in a given tank with a floating roof at the acquisition time of each image, the vertical position of the roof in the first image must be determined. If the floating roof moved at some point during the period covered by the time series, its vertical displacements have already been estimated in the previous step, and therefore only its initial position is missing. If the roof did not move, the displacements estimated in the previous step will be invalid, but the initial position of the floating roof will be all of the information that is required anyway. Here, a simple decision rule will be established to determine whether a floating roof has moved or not, and a method to estimate its initial position will be described for both of these scenarios.

If many CSs in the image patch containing a storage tank moved, this will imply that its floating roof has very likely moved. The weights computed in the previous step when estimating the vertical displacements of the roof provide a good indication for this: each weight w_{ij} quantifies the maximum number of moving CSs matching between the corresponding image pair. The average of all these weights, denoted as w_{avg}, can be used as a simple metric to quantify the number of CSs which could potentially correspond to a moving roof.

If the floating roof did not move, there will be very few CSs which moved, and the value of w_{avg} will be low. Additionally, the floating roof (and therefore its semicircular double reflection) will be present in the binary image S which shows the CSs which remained static. Therefore, if many static CSs match a semicircle towards far-range, it will indicate that the roof did not move. To obtain the number of static CSs which match this semicircle, the method previously

described in Section 4.2.3 can be used. Rather than applying this method to the CSs detected in a single image, in this case it should be applied to the static CSs in S. Now, the function with the cross-correlation values for each layover l can be denoted as $f_s(l)$, and the layover l_{r_s} for which the highest number of static CSs match such a semicircle can be obtained using equation 4.13.

The values of w_{avg} and $f_s(l_{r_s})$ can now be compared: if $f_s(l_{r_s}) > w_{avg}$ the floating roof is assumed to have remained static, and its layover will be given by l_{r_s}, which can be converted into the corresponding height using equation 4.4. On the other hand, if $w_{avg} \geq f_s(l_{r_s})$ the roof is assumed to have moved. The vertical displacements of the floating roof have been obtained in the previous step, but its initial position still needs to be determined.

The method described in Section 4.2.3 can be applied to any of the n images showing the CSs which moved, denoted as M_i (with $i = 1, ..., n$), to estimate the position of the floating roof at the corresponding acquisition time. The initial position of the roof could therefore be simply estimated using the first image (M_1). However, a more accurate and robust estimation is possible by jointly using all of these n images: because the displacements of the roof are known, the initial position can be obtained from each of these n images by subtracting the displacements accordingly. The functions with the cross-correlation values obtained from each of the n M_i images, denoted here as $f_{m_i}(l)$ (with $i = 1, ..., n$), can be combined to estimate the layover l_{r_1} due to the initial position of the floating roof:

$$l_{r_1} = \underset{l \in [0, l_r^{max}]}{\arg\max} \sum_{i=1}^{n} f_{m_i}(l - d'_{1i}) \tag{4.20}$$

Here, d'_{1i} denotes the vertical displacement of the floating roof between the first image and image i, which can be computed as:

$$d'_{1i} = \begin{cases} 0 & \text{if } i = 1 \\ \sum_{k=1}^{i-1} d'_k & \text{if } i > 1 \end{cases} \tag{4.21}$$

When evaluating $f_{m_i}(l - d'_{1i})$ in equation 4.20, it is important to note that the functions $f_{m_i}(l)$ have been computed using cross-correlation only inside the valid interval $l \in [0, l_r^{max}]$, and will have a value of 0 outside of this interval.

Finally, the amount of oil stored inside the tank can then be computed for each of the images in the series. If the floating roof did not move, it will have a constant layover value l_{r_s}. On the other hand, if it moved its layover for any given image i in the series can be expressed as: $l_{r_i} = l_{r_1} + d'_{1i}$. In both cases, these layover values must be converted into the corresponding heights, and the volume of oil stored can then be computed using equation 4.6.

4.3.5 Classification of storage tank type

In Section 4.2.4 a method was introduced to automatically determine whether a given storage tank has a floating or a fixed roof. This method uses a machine learning classifier trained with three features obtained when estimating the dimensions of a storage tank from a single SAR image. These three features represent the number of CSs which match the two detected semicircular double reflections towards near-range (present in both types of tanks), and the one towards far-range (present only in tanks with a floating roof).

When using a SAR time series the classification approach will be similar, but the available temporal information can now also be used to achieve a better classification performance. If a floating roof moves at some point during the time series, the corresponding CSs will move as well, whereas on a fixed roof all or most of the CSs will remain static. Therefore, the value of w_{avg} (defined in Section 4.3.4) will allow to very easily identify tanks with a floating roof if their roofs

move at some point during the time series. If a floating roof did not move, the value of w_{avg} will be low, but the value of $f_s(l_{r_s})$ (also defined in Section 4.3.4) can be used instead. The maximum of these two values can therefore be used as a feature for the classifier, for which a high value will indicate the presence of a floating roof, and a low value a fixed roof. Finally, similarly to the single image case, the number of static CSs matching the double reflections at the bottom and the top of a storage tank, given by $h(x_b, y_b, r_t)$ and $h(x_b, y_b - l_t, r_t)$, will also be used as the two final features. For these two last features, a small difference with the single image case is that now the Hough accumulator $h(x, y, r)$ was computed using only the CSs which remain static during the series. In summary, when estimating oil storage using a time series, the following three dimensional feature vector v can be generated for each storage tank:

$$v = \left(\max\left(w_{avg}, f_s(l_{r_s})\right), h(x_b, y_b, r_t), h(x_b, y_b - l_t, r_t)\right) \tag{4.22}$$

As in the single image case, the use of a small number of features will imply that just a few training samples suffice to train an accurate classifier.

5 SAR change detection using coherent scatterers

This Chapter presents a new unsupervised CD method for pairs or series of VHR SAR images, named Change Detection by Coherent Scatterers (CDCS). This method will detect changes caused by the appearance, disappearance, or movement of man-made objects inside the imaged scene, as well as changes to static objects, while ignoring changes to natural targets such as vegetation. Part of the material in this Chapter has been published in [Villamil Lopez & Stilla, 2022].

This Chapter is organized as follows: initially, Section 5.1 describes how different types of changes between a pair of SAR images can be detected by coherently comparing the CSs detected in the two images. An extension of this method for longer time series is described in Section 5.2. Here, a new CD metric is introduced, which exploits the full coherence matrix and is able to ignore irrelevant transient changes. Because the changes to be detected are often significantly larger than individual CSs, the pixelwise CD using CSs is followed by an object-based change analysis step, which is described in Section 5.3. This consists of a clustering step, followed by the segmentation of the changed objects. Finally, certain types of changes can be identified by their size and/or temporal behavior. A block diagram of the complete processing chain can be seen in Fig. 5.1.

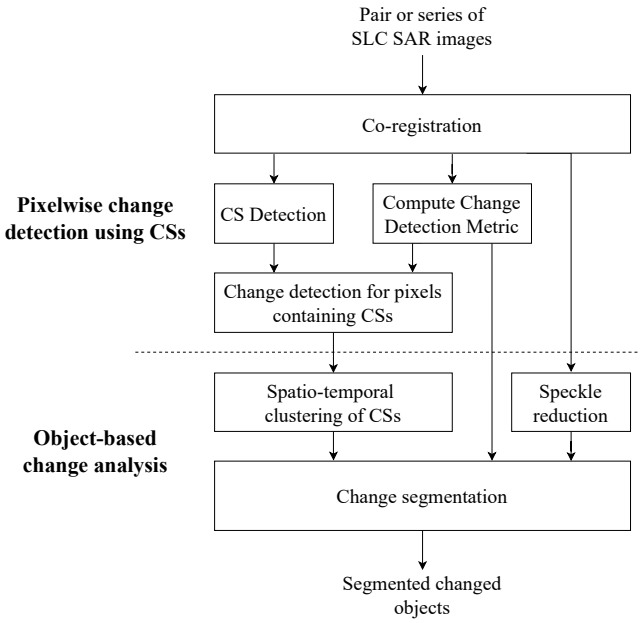


Figure 5.1: Processing chain of the proposed change detection approach.

5.1 Change detection on an image pair using CSs

In Section 4.3.1, CCD was applied to determine which CSs moved and which remained static over a series of images acquired at different times. This same principle will be extended here and developed into a general CD method capable of distinguishing different types of changes associated to man-made objects. For completeness, the basic principle will be briefly described here once again.

If a CS experiences even a small change (e.g., a subpixel displacement) the interferometric coherence of the corresponding pixel will drop significantly. On the other hand, if this CS remains unchanged and static the coherence will have a high value, as strong point scatterers typically have high coherence and are not significantly affected by temporal decorrelation [Ferretti et al., 2001; Eineder et al., 2009]. To implement such a CD method, the two SLC SAR images should first be co-registered with subpixel accuracy, and the CSs should be detected for each image as described in Section 3.2. The pixels of the resulting two binary images get the value 0 or 1 depending on the absence or presence of a CS. Absence of a CS is assumed to be clutter. Additionally, the interferometric coherence should be computed for the image pair, as it will be used as the CD metric. A threshold γ_t should then be applied to the resulting coherence image in order to detect which of these CSs have changed or moved during the interval between the two image acquisitions. Finally, depending on the results of the CS detection and the coherence thresholding, different types of changes can be distinguished. The possible cases are summarized in Table 5.1 and described below.

If a pixel containing a CS has a coherence below γ_t, it will imply that a change involving a man-made object has very likely occurred there. Three different types of changes can then be distinguished. If a CS was only detected in the first (i.e., earlier) image, it will imply that an object left the scene, whereas if it was detected only in the second image, it will imply that a new object appeared. If a CS was detected in both images, then either one object changed or moved, or it was replaced by a different object. On the other hand, if the coherence of a pixel containing a CS is higher than γ_t, it will be considered that this CS remained unchanged and is present in both images, even if it was only detected in one. The reasoning behind this is that a high coherence value should only be possible if there is no significant change, whereas false negatives in the CS detection are much more likely to occur.

Finally, the pixels containing clutter in both images are ignored, independently of the change in amplitude or coherence loss. These pixels could correspond to an environmental change, and the focus of the proposed method is to detect only those changes associated with man-made objects. Besides, the proposed CCD metric would not work well for these pixels, as it assumes

Table 5.1: Interpretation of the change detection results.

Image 1	Image 2	Coherence	Type of change
Clutter	Clutter	-	Irrelevant
CS	Clutter	Low	Object disappeared
Clutter	CS	Low	Object appeared
CS	CS	Low	Different object or object changed/moved
CS	CS	High	Same object (no change)
CS	Clutter	High	Same object (no change)
Clutter	CS	High	Same object (no change)

that unchanged pixels have high coherence values. The distributed scatterers present in natural targets are affected by temporal decorrelation. This is especially relevant at X-band [Parizzi et al., 2009], which is used by most high resolution spaceborne SAR sensors. In contrast, CCD can be robustly applied to CSs even with long temporal baselines.

When applying this method, the computed coherence image should have the same pixel spacing as the original SLC images, independently of the number of looks used to estimate the coherence and the associated resolution loss. The coherence values are only evaluated on the pixels containing a CS in at least one image. Because the images with the detected CSs have the full resolution, the window size used for the coherence estimation does not affect the spatial resolution of the results. The effect of the window size used for the coherence estimation will be analyzed later in the experiments shown in Chapter 7.

5.2 Change detection on a time series using CSs

If a time series with more than two repeat-pass images is available, the proposed CD method can be extended. Instead of detecting changes between each of the consecutive image pairs separately, the CD task can be formulated in a slightly more general way. The goal is now to estimate the time interval $[t_{start}, t_{end}]$ when each of the detected CSs is present in the imaged scene and remains unchanged. Using CD, the exact values of t_{start} and t_{end} for a given CS cannot be determined, but these can be narrowed down to the following intervals: $t_{start} \in (t_{a-1}, t_a)$ and $t_{end} \in (t_b, t_{b+1})$. Here, a and b are the indices of the first and last images where that specific CS was detected, and t_i denotes the acquisition time of the i-th image of the series. When using a time series with n images, up to n different CSs can be detected in a given pixel, each present during a different time interval. A higher temporal resolution of the time series allows to detect more and faster changes, and better determine the change occurrence.

When performing CD with a time series, instead of simply comparing each image to the next one, all the image pairs in the series can be compared. This can be done using the coherence matrix for each pixel. This matrix contains the coherence value for images j and k (denoted as $\gamma_{j,k}$) at row j and column k. The additional coherence values can be exploited to distinguish relevant changes to an object (e.g., an object appears, leaves the scene, moves, or is modified in a lasting manner) from irrelevant transient changes. Here, transient changes are defined as those situations where an object is just temporarily affected by some external factor in one or a few outlier images and does not actually change. Possible examples are an object temporarily covered with snow or occluded due to the radar shadow casted by another object. One could argue that these situations are actual changes that should be detected, but they are typically irrelevant for many practical applications. Therefore, the ability to distinguish transient changes (e.g., a building temporarily covered with snow) from those where the object itself changes (e.g., a new building is built or an existing building is renovated or demolished) represents a clear advantage. Later in Section 7.2, a real example of a transient change is shown and compared to another example where an object itself actually changes. Both are illustrated by their characteristic coherence matrices.

To determine whether a change happened to a given CS during the time interval (t_i, t_{i+1}), all the image pairs with one image acquired at t_i or earlier and another at t_{i+1} or later can be taken into account. When a man-made object experiences a significant change, the coherence should be low for all these image pairs. On the other hand, for transient changes, the coherence is low for the pairs containing one of the outlier images, but high for other image pairs with longer temporal baselines. This indicates some event briefly affected the object causing its coherence to drop, but the object later returned to its exact previous state and therefore it did not actually change.

Based on this, a new CD metric f_i can be defined to determine whether a significant change happened to a given CS during the time interval (t_i, t_{i+1}). For the case of CD with an image pair introduced in Section 5.1, this metric was simply the coherence of the pair: $f_i = \gamma_{i,i+1}$. In this case, the new metric should be insensitive to transient changes. This can be achieved by exploiting additional image pairs with different temporal baselines, enforcing that the coherence must be low for all of them in order to detect a change. This new metric can be computed as follows:

$$f_i = \max_{\substack{j,k \\ j \leq i \\ k \geq i+1}} \gamma_{j,k} \tag{5.1}$$

As this metric computes the maximum of many coherence images for each pixel, it is important to avoid noisy coherence estimates. For this, a large window can be used during the coherence computation. Storing the coherence matrix for every pixel requires lots of memory, especially for long time series. However, this is not required: it is enough to initialize $n-1$ empty images to store the values of f_i. After computing the coherence for a given image pair, the values of the images for f_i which include that pair can be updated accordingly. This is more memory efficient, but the coherence still needs to be computed for $(n^2-n)/2$ image pairs. As a small subset of the coherence matrix suffices to identify transient changes, the metric f_i from equation 5.1 can be slightly modified to reduce the computational cost:

$$f_i = \max_{\substack{j,k \\ j \in [i-r,i] \\ k \in [i+1,i+1+r]}} \gamma_{j,k} \tag{5.2}$$

where r can be used to increase the number of elements of the coherence matrix to be taken into account. A value of $r = 1$ should be enough to detect transient changes affecting only one image, whereas a value of $r = 0$ would reduce this metric to the previous one used for an image pair: $f_i = \gamma_{i,i+1}$.

To perform CD with a time series using this new metric, the CS detection should be performed for all the co-registered SLC images as described in Section 3.2.1. This results in a stack of binary images C_i (with i in $1, \ldots, n$) with the detected CSs for each image. Additionally, the CD metric should be computed using equation 5.2, resulting in a stack of images with the values of f_i for each pixel, with i in $1, \ldots, n-1$. The remaining steps can be performed exclusively for the list of pixels containing CSs instead of using the full image raster. This reduces the required memory and makes the computations faster. For this, a binary mask showing the pixels with a CS in at least one image can be computed: $C_{max} = \max_i C_i$.

As introduced in Section 5.1, some pixels might exhibit inconsistencies, with the CD metric indicating that no change occurred between two images but the CS detection indicating that only one contains a CS (i.e., $f_i \geq \gamma_t$ and $C_i \neq C_{i+1}$). Before, it was argued that the most likely cause for this are false negatives in the CS detection. The proposed solution was to consider that an unchanged CS is present in both images, effectively correcting the value of C_i or C_{i+1}. The same principle can be applied now, but the comparison should be carried over across the whole series (e.g., a CS detected in just one image could be present in several images before and after). This consistency check can be implemented with a forward pass (sequentially comparing images i and $i+1$, with $i = 1, \ldots, n-1$), followed by a backward pass (comparing images i and $i-1$, with $i = n, \ldots, 2$). This results in corrected values for the CS detection, denoted as C'_i. These will have the same values as C_i except for the likely false negatives corrected, where $C'_i = 1$ and $C_i = 0$.

After this consistency check the CD can finally be performed. First, the unchanged and static CSs can be identified by finding the pixels where f_i is always above the threshold: $\min_i f_i \geq \gamma_t$.

Each of these pixels contains a single CS with $t_{start} < t_1$ and $t_{end} > t_n$. For the remaining pixels, the number of different CSs and the time periods during which these are present need to be determined. For a given pixel, it can be established that a new CS appeared at a time $t_{start} \in (t_{i-1}, t_i)$, with $i > 1$, if a CS is present in image i and a change happened in between images $i-1$ and i (i.e., if $C'_i = 1$ and $f_{i-1} < \gamma_t$). Additionally, CSs present in the first image ($C'_1 = 1$) appeared before the start of the series: $t_{start} < t_1$. In a similar way, it can be established that a CS disappeared at a time $t_{end} \in (t_i, t_{i+1})$, with $i < n$, if if $C'_i = 1$ and $f_i < \gamma_t$. Also, those with $C'_n = 1$ are still present at the end of the series: $t_{end} > t_n$. After checking these conditions for all the pixels and all images, a list of detected CSs (each with an estimated time period) is obtained for each pixel.

The proposed consistency check does not account for false positives in the CS detection. These are unlikely and usually have a negligible effect, resulting in a few wrongly detected CSs which often appear isolated, spread across the imaged scene. In contrast, the much higher number of correctly detected CSs appear in the areas where man-made objects are located. However, false positives can pose a problem in certain areas that remain unchanged and are unaffected by temporal decorrelation. At these locations, the proposed consistency check would propagate the CSs wrongly detected in each image to other images of the series, resulting in a higher number of false positives. This issue becomes more significant with longer time series, but it can be mitigated by a simple post-processing step. If the CD determines that a given CS is present in images a through b (both included), the corresponding values of C_i can be checked to see in how many images it was originally detected. If this number is too low, then the corresponding CS was likely a false positive and can be discarded:

$$\sum_{i=a}^{b} C_i < k(b-a+1) \Rightarrow \text{ discard CS, likely false positive} \tag{5.3}$$

with k being a value between 0 and 1 which controls the maximum fraction of false negatives to be corrected by the consistency check.

5.3 Segmentation and analysis of the detected changes

5.3.1 Spatio-temporal clustering of CSs

A clustering algorithm can be applied to detect objects from a set of point scatterers, as man-made objects appear in SAR images as clusters of densely packed CSs. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [Ester et al., 1996] seems well suited for this task. Given a set of points, DBSCAN clusters together those closely packed in high-density regions. Isolated points in low-density regions are marked as outliers, which makes the algorithm robust against noise. DBSCAN can detect clusters of arbitrary shapes and its definition of a cluster aligns well with this particular problem and data. Its robustness against noise can discard most of the false positives in the CS detection. Besides, the number of clusters (unknown in this case) does not need to be specified.

The DBSCAN algorithm has two parameters: a radius ε, and p, the minimum number of points within this radius required to form a cluster. The metric used to compute the distance between points can also be specified. For this use case, the distance in meters between two pixels is used. This makes the selection of the radius ε more intuitive and compensates different pixel spacings in azimuth and range. This distance is computed by scaling the pixel coordinates of each CS by the pixel spacing along the corresponding axes. For the range axis, typically in slant-range projection, the equivalent pixel spacing in ground-range is computed using the mean incidence angle.

The values of ε and p should simply control the density of CSs required to form a cluster. Constraints related to the minimum cluster size can be imposed later by evaluating the number of CSs and/or the area inside the convex hull of the resulting clusters. Otherwise, high values could be selected for both parameters, making it difficult to correctly cluster objects of arbitrary shapes and increasing the probability of grouping nearby objects into the same cluster.

The described method can be used to perform spatial clustering on a set of CSs. However, the temporal information obtained from the CD can also be considered for the clustering. All the CSs belonging to the same object are expected to appear and disappear at the same time. This can be enforced by splitting all the CSs into multiple subsets according to the different time intervals (i.e., the combinations of t_{start} and t_{end}). The DBSCAN algorithm can then be applied separately to each of these smaller point sets. This way, CSs closely located in an image but belonging to different objects can be separated if they have a different t_{start} or t_{end}. Besides, it also makes the clustering significantly faster, as most DBSCAN implementations run on quadratic time.

5.3.2 Segmentation of the detected changes

The previous clustering step grouped the CSs which are likely to belong to the same objects. However, it would be desirable to obtain a dense change map for each object. For this, each changed object should be segmented using all the image pixels and not only CSs. The segmentation is performed separately for each cluster using all the relevant images, as t_{start} and t_{end} are known. The extent of the image patch to be processed can be obtained by getting the rectangle enclosing the cluster's convex hull. This extent can be increased by some scaling factor as a safety margin, in case the cluster is smaller than the corresponding change to be segmented.

For objects present in more than one image, the CD metric f_i can be thresholded to obtain an accurate segmentation of the changes. If an object was first seen at image a and last seen at image b, its pixels must:

- change in the interval (t_{a-1}, t_a): $f_{a-1} < \gamma_t$
- change in the interval (t_b, t_{b+1}): $f_b < \gamma_t$
- remain unchanged in the interval $[t_a, t_b]$: $\min\limits_{a \leq i < b} f_i \geq \gamma_t$

These constraints result in three binary images that can be combined using a logical AND operation to obtain a change mask. This mask contains all the pixels inside the cluster and near its boundaries exhibiting the same temporal behavior. In the case that $a = 1$, it will not be possible to apply the first constraint, whereas if $b = n$, the same will happen to the second constraint. However, in both cases the two remaining constraints are sufficient.

For objects present in only one image (i.e., if $a = b$) it will not be possible to apply the third constraint. The other two might not suffice to segment the corresponding change, as the object surroundings might also have low coherence due to temporal decorrelation. In this case, some additional constraints related to the SAR amplitude should be added:

- amplitude change in (t_{a-1}, t_a): $|A_a - A_{a-1}| \geq \Delta_A$
- amplitude change in (t_a, t_{a+1}): $|A_a - A_{a+1}| \geq \Delta_A$
- amplitude should not have low values at t_a: $A_a \geq A_{min}$

A_i denotes the SAR amplitude in dB scale of the image i in the series. The parameter Δ_A represents the minimum amplitude difference in dB to consider that a pixel changed. This is equivalent to the well-known log-ratio metric for ICD [Bazi et al., 2006]. While there are more advanced metrics, this fixed threshold should already perform well, as it is only applied locally where changes have already been detected. The second parameter A_{min} represents the minimum amplitude to consider that a pixel might belong to a man-made object, as these typically exhibit high amplitude values. Before applying these new constraints, the speckle noise should be reduced by applying multilooking or a more sophisticated speckle filter. Some modern methods [Deledalle et al., 2015; Dalsasso et al., 2022] achieve a very good denoising performance and preserve the full spatial resolution. These constraints only work well if the used SAR sensor is well calibrated, which is typically the case for modern spaceborne SAR sensors.

Finally, the change mask obtained for each cluster is refined by applying mathematical morphology operations. First, a closing operation with a radius of a few pixels is used to fill small holes in the mask without significantly changing its shape or size. Then, the connected components that are too small or mostly outside of the cluster's bounds are discarded. The remaining connected components should provide a good segmentation of the changed object (or objects) in the cluster.

5.3.3 Spatio-temporal analysis of the segmented changes

To detect changes corresponding to specific events, the obtained results can be analyzed to identify objects with certain temporal behaviors and of certain sizes. This is especially interesting for urban areas and similarly complex scenes where many changes occur between two consecutive image acquisitions. To focus on objects above or below certain sizes, a threshold can be applied to the area of the segmented changes. Changes can also be categorized according to their duration (e.g., into fast, long-term and permanent changes). Also, further constraints regarding their time of appearance and disappearance can be imposed. Below, a few examples are provided, illustrating how this analysis can be applied.

Newly constructed or renovated buildings and infrastructure typically imply the appearance of new CSs which later remain unchanged over a long time. Such changes can be identified by imposing the following constraints: $t_{start} > t_1$, $t_{end} > t_n$ and $t_{end} - t_{start} > \Delta T$. This requires at least 3 images: one acquired before the construction work is finished, and two acquired afterwards and at least ΔT time apart. Here, ΔT is set to the minimum amount of time that an object must remain unchanged to be considered a new static object (e.g., a couple of months).

Moving objects (e.g., parked cars, airplanes in airports, etc.) typically imply CSs appearing and disappearing inside short periods: $t_{end} - t_{start} < \Delta T$. Again, at least 3 images are required to unambiguously identify this behavior: one acquired before the object appears, one with it present, and one after it leaves. Here, ΔT is set to the maximum amount of time that moving objects are expected to remain static (e.g., from multiple days to a couple of weeks).

For this kind of temporal analysis, it is important to note that t_{start} and t_{end} (and therefore also the duration) can only be narrowed down to some intervals, as introduced in Section 5.2. In this work, the threshold on this time length is applied to the lower bound of this interval. However, this could be handled differently depending on the application.

6 Object recognition with a fully convolutional Siamese network

This Chapter introduces a new method for object recognition in a single VHR SAR image, a task also known as SAR automatic target recognition (ATR). Instead of using one of the typical network architectures for object detection, the task is formulated as a template matching problem that will be solved using a fully convolutional Siamese network architecture. This type of network is mostly used for visual tracking, but it will be adapted here for SAR ATR. By creating a template database with a set of representative training samples of the different objects to be detected, these objects can then be detected in new images during inference. The proposed network can be trained with relatively few labelled samples without overfitting. This kind of architecture is also well suited for few-shot or even one-shot learning. To the author's best knowledge, this is the first time that a fully convolutional Siamese network has been applied for SAR ATR. In the near future, part of the material in this Chapter will be submitted to a journal for publication.

This Chapter is organized as follows: initially, the challenges of object recognition in SAR images are briefly analyzed in Section 6.1. Based on this, a suitable network architecture is selected and described in Section 6.2. For the reader's convenience, the idea behind the chosen network architecture is also briefly explained here, together with the modifications implemented to adapt it for the task of SAR ATR. Section 6.3 describes the proposed training strategy for the chosen network architecture. Finally, Section 6.4 explains how the trained network can be used for inference (i.e., to detect and classify objects in new SAR images).

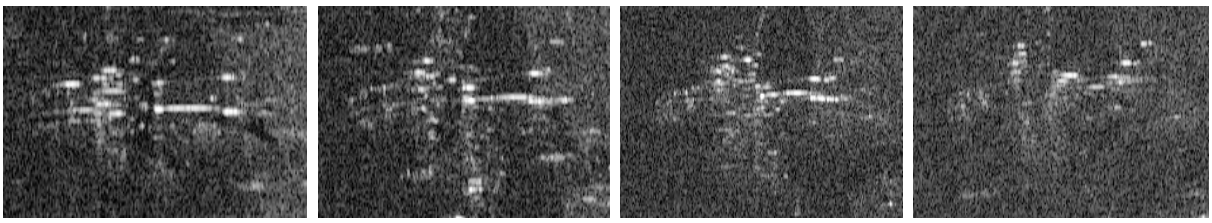
6.1 Challenges of object recognition in SAR images

In recent years there has been great progress in the field of SAR ATR. Nevertheless, optical imagery is still more widely used for object detection tasks, with several private companies offering mature commercial solutions. This is in part due to the lack of high quality open datasets with VHR SAR data, as well as the higher difficulty of accurately labelling objects in SAR images. Besides, there are also some challenges specific to SAR images which do not apply to nadir optical imagery. In spite of this, the same methods used for optical images are often applied to SAR images with only minimal changes. Here, these challenges specific to SAR images will be briefly analyzed, as they will be taken into account afterwards when choosing a network architecture. These challenges will be illustrated with real examples of airplanes in VHR TerraSAR-X images, part of a dataset which will be described later in detail in Chapter 7. These images were transformed for visualization as described in Section 3.1.

The main challenge of SAR ATR is inherent to the imaging principle: in SAR images, man-made objects tend to appear as sets of discrete point scatterers surrounded by clutter, and the whole image is affected by speckle noise. Also, even though an object's radar shadow can give valuable information on its geometry, this shadow can only be clearly seen in certain conditions.

(a) (b)

Figure 6.1: Appearance of the same airplane on a VHR SAR and optical images: a) SAR, b) optical (taken from Google Maps).



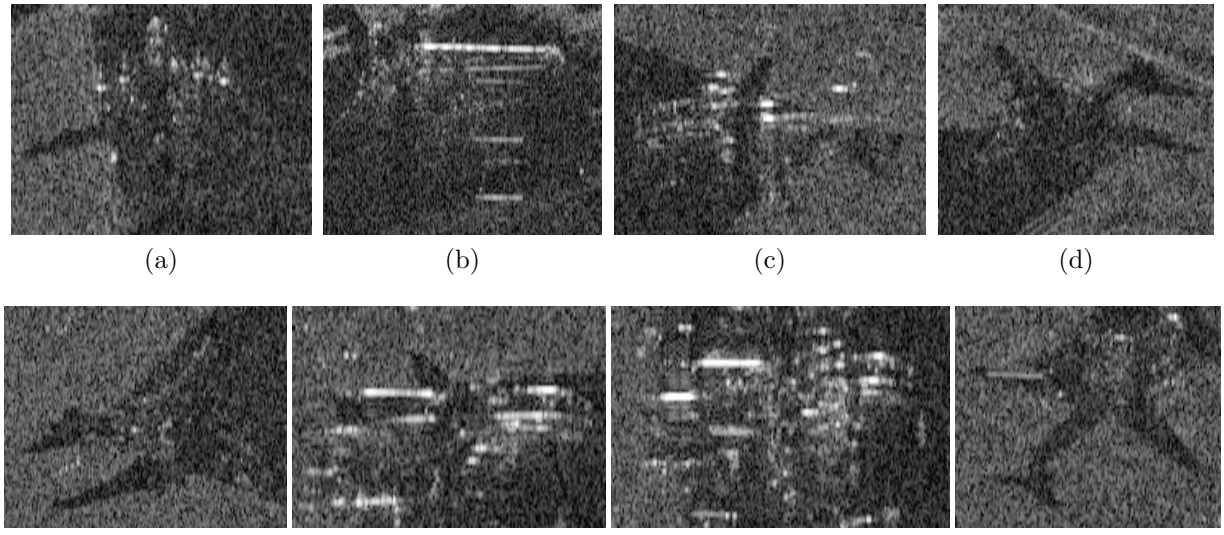
(a) (b) (c) (d)

Figure 6.2: Appearance of the same airplane for different incidence angles: a) 23°, b) 36°, c) 54°, d) 60°.

On the other hand, in optical images objects typically appear as several continuous and homogeneous image regions delimited by sharp and clear edges. Because of this, it is typically much more difficult to differentiate man-made objects from the background in SAR images than in optical images. This is illustrated in Fig. 6.1, where the same airplane is shown in two images (one SAR, one optical) of similar resolution. In the SAR image only the backscattering from the airplane's four engines and its cockpit can be clearly seen. Besides, only the radar shadow casted by the left wing is clearly visible, as this wing is located above grass, whereas the rest of the aircraft is located above asphalt. In contrast, in the optical image all the imaged parts of the airplane can be clearly seen, as well as its shadow.

Due to the side-looking geometry and range-based imaging of SAR sensors, a variation of the imaging geometry will also have a strong effect in the appearance of man-made objects in SAR images. The effect of different incidence angles can be seen in Fig. 6.2. These four images show the same type of airplane, at the same location, imaged from the same direction but with four different incidence angles. As it can be seen, the SAR signature of a given object changes significantly for large variations in the incidence angle. Imaging with a different incidence angle also changes the amount of backscattered energy, which further alters the overall characteristics of the SAR image. This can also be seen in Fig. 6.2: steep incidence angles (e.g., 23°) result in an image with much better contrast (e.g., the radar shadow is partially visible even on asphalt) than shallow angles (e.g., 60°).

Changes in the orientation of an object with respect to the radar's line of sight will have an even more significant effect than changes to the incidence angle. This can be seen in Fig. 6.3, which shows eight airplanes of the same type but with different orientations, all taken from the same SAR image. This illustrates how the SAR signature of this airplane completely changes for the different orientations. Even relatively small rotations can introduce significant changes (e.g., due to very strong double reflections occurring only at specific angles), as can be seen when comparing Fig. 6.3a and 6.3b. Also, certain orientations result in almost no backscattering from



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

(e) (f) (g) (h)

Figure 6.3: Appearance of the same airplane for different orientations with respect to the SAR sensor.

(a) (b)

Figure 6.4: Appearance of the same airplanes on different seasons: a) summer, b) winter (with snow).

the airplane (e.g., Fig. 6.3d and 6.3e), which can then only be clearly identified by its radar shadow (if visible).

An important advantage of SAR images is that the appearance of the imaged scenes is not altered by changes in sunlight illumination, clouds, or similar atmospheric effects. However, seasonal effects such as snow can also greatly alter the appearance of man-made objects in SAR images. This behavior can be seen in Fig. 6.4, which shows three airplanes of the same class imaged at the same location at two different times, using the same SAR sensor and imaging geometry. The image in Fig. 6.4a was acquired during summer and the one in Fig. 6.4b during winter, with the runway partly covered with snow. In Fig. 6.4b, two different effects caused by the snow can be observed. The appearance of the airplanes at the top and bottom are altered by the fact that they stand on a different surface (i.e., snow instead of asphalt). The appearance of the airplane in the center is even more different as the airplane itself is covered with snow.

6.2 Network architecture

The aforementioned challenges should be taken into account when developing a SAR ATR method. Some can be addressed during pre-processing (e.g., the speckle noise can be greatly reduced by applying despeckling techniques), but others will influence the selection of a network architecture. Due to the lack of large open datasets and the difficulty of accurately labelling SAR data, it should ideally be possible to train the chosen network with relatively few labelled samples without overfitting. To make the method flexible, it should ideally also be capable of few-shot or even one-shot learning: given a network trained on abundant data for some object classes, it should be able to detect novel classes based on just a few examples, or even a single one. This is also important as the data can often be imbalanced, with only a few samples available for the less common object classes. As these classes naturally appear less often, gathering more samples is not always feasible.

In order to achieve this, instead of using a traditional classification or object detection network, similarity learning will be applied by using a deep Siamese CNN. Siamese networks take two inputs and process them separately using the same NN, and then apply another function to combine these two outputs. Often, a simple distance or similarity metric is used for this, and Siamese networks are trained to evaluate the similarity of their two inputs. With such a network, classification can be performed by comparing an unknown input with some examples from all the known classes to find the one with the highest similarity. This kind of network architecture can be directly used to perform one-shot learning: a single sample of a new object class can be used as one of the inputs to detect more instances of this object, without any additional training. Also, as these networks take a par of inputs and the number of pairs in a given dataset increases quadratically with respect to the number of annotated samples, a relatively low number of samples will lead to many different inputs for training such a network, which should make overfitting more difficult. Among other tasks, Siamese CNNs have been used for one-shot character recognition [Koch et al., 2015] or face verification [Taigman et al., 2014; Schroff et al., 2015]. More recently, Pan et al. [2019] applied a Siamese CNN to the MSTAR dataset for target classification in SAR images, achieving a good performance with very few training samples.

Using a Siamese network also makes it possible to treat instances of the same object class but imaged with different geometries as different classes. While this would not make much sense for nadir optical imagery, in SAR images a different imaging geometry can induce very large differences, to the point that the same object can look completely different (e.g., as shown in Fig. 6.3). In some cases, two different objects imaged from the same direction can appear more similar than two objects of the same type imaged from different directions. Treating different imaging geometries for a given object as different classes would quickly result in too many classes for a classification network, but can be easily implemented with a Siamese network by choosing a suitable similarity criterion (i.e., two objects will only be considered similar if they are of the same class and were imaged with similar incidence angle and orientation with respect to the SAR sensor). The imaging geometries of two objects are considered similar if the two following conditions are fulfilled:

$$|\theta_1 - \theta_2| \leq \delta_\theta \tag{6.1}$$

$$\min\left(|\alpha_1 - \alpha_2|, 360 - |\alpha_1 - \alpha_2|\right) \leq \delta_\alpha \tag{6.2}$$

where θ_1 and θ_2 denote the two incidence angles, and α_1 and α_2 the object's orientations. δ_θ and δ_α are two hyperparameters which can be used to modify the tolerance of the similarity criterion. As a side benefit, this similarity criterion allows to estimate the approximate orientation of objects in addition to their classes. However, it requires some additional information to be available for each annotated object: the incidence angle (which can be automatically obtained from the

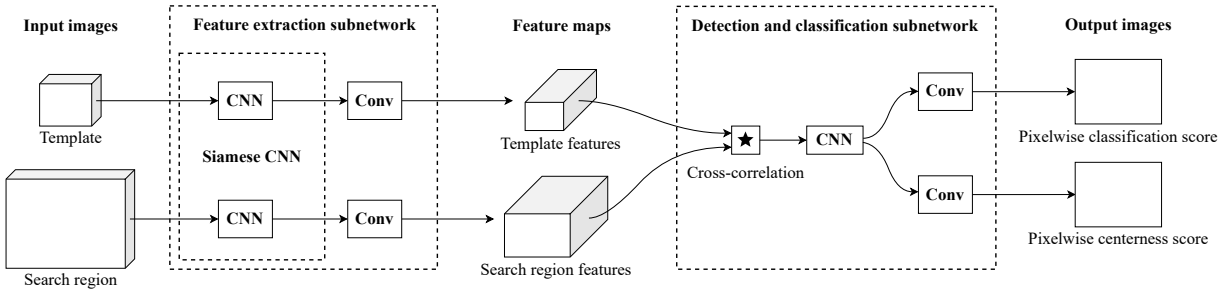


Figure 6.5: Architecture of the fully convolutional Siamese network used for object detection.

metadata of the corresponding SAR image) and the approximate orientation angle, which needs to be manually annotated (e.g., by using rotated bounding boxes aligned with a certain axis of the object).

Conventional Siamese networks take two inputs of the same size. However, the aim here is to perform both object detection and classification, locating multiple instances of different types of objects in a SAR image. Using similarity learning, this can be formulated as a template matching problem: given an exemplar SAR image patch of an object, other instances of this object should be located in a much larger search image. This can be achieved using a Siamese fully convolutional (SiamFC) network. This network architecture was introduced by Bertinetto et al. [2016] for the task of tracking arbitrary objects in videos, which is essentially a template matching problem. By making the network fully convolutional (and therefore translation invariant) and using cross-correlation to combine the feature maps of the two inputs, the similarity function can be computed in a single evaluation for all translated sub-windows within the search image. This is mathematically equivalent to evaluating a regular Siamese network on each translated sub-window separately and combining the feature maps using the inner product. However, it is much more efficient, and it is both advantageous during training and inference. This type of network architecture has become very popular for visual tracking due to its very good performance and its ability to operate in real-time. In recent years, many improvements upon the original SiamFC architecture have been introduced [Li et al., 2018, 2019a; Guo et al., 2020; Xu et al., 2020]. In this chapter, a network architecture based on the more recent SiamFC++ network by Xu et al. [2020] will be modified for the task of object detection and classification in VHR SAR images.

The chosen network architecture can be seen in Figure 6.5. In this diagram, each Conv block stands for a single convolutional layer, whereas each CNN block stands for a deep network with multiple layers. Even though this network is trained end-to-end, two different subnetworks can be distinguished. The feature extraction subnetwork computes the feature maps for both the template and the search image. Then, the detection and classification subnetwork combines both feature maps using cross-correlation and applies some additional post-processing to locate the template in the larger search image. Below, these two subnetworks will be described in detail.

The feature extraction subnetwork is comprised by a Siamese CNN (i.e., two identical networks with shared weights to process the two inputs) followed by two non-shared convolutional layers (i.e., with different weights for each input branch), as suggested by Xu et al. [2020]. In the original SiamFC paper, Bertinetto et al. [2016] use a network similar to the convolutional stage of the relatively shallow AlexNet [Krizhevsky et al., 2012] for the Siamese CNN. Deeper networks typically use padding, which breaks translational invariance. Li et al. [2019a] have shown that the spatial bias introduced when using deeper networks with padding can be addressed during training by applying random translations to the input data. This enables the use of deeper CNNs, such as the very popular ResNet [He et al., 2016] or the more recent ConvNeXt [Liu et al., 2022].

Therefore, the selection of this CNN will be left open, and different backbone networks will be compared later during the experiments. The two Conv layers following this Siamese CNN have a 3×3 kernel size, the same number of output channels as the preceding CNN, a stride of 1 and no padding. BatchNorm [Ioffe & Szegedy, 2015] is applied directly after each of these convolutional layers, and no additional activation function (e.g., ReLu) is used. The output of this subnetwork are two feature maps of different sizes which will be combined by the subsequent detection and classification subnetwork.

For adapting a SiamFC network for SAR ATR, a modification pertaining the BatchNorm layers in the Siamese CNN needs to be implemented. These layers, which are used in many popular CNNs, normalize their inputs to make training faster and more stable [Ioffe & Szegedy, 2015]. BatchNorm behaves differently during training and inference. During training, each batch is normalized using its own statistics. During inference, the inputs are normalized using statistics learnt during training from the complete training dataset. For the proposed application, the statistics of the two input SAR images will typically be very different: the template will always be a small image patch containing a single man-made object, whereas the search image will be much larger and often contain clutter areas (e.g., natural targets like vegetation). Because of this, during inference BatchNorm should ideally be performed using different statistics for the two branches of the Siamese CNN. The two branches of the Siamese CNN will therefore share all the layers and weights except the BatchNorm layers, which will be different for each branch (so that they can learn the statistics of their own inputs). The importance of this modification can be empirically verified, as it will be shown later during the experiments.

The detection and classification subnetwork takes the two feature maps as inputs and combines them using a cross-correlation operation. The cross-correlation is performed channel by channel (i.e., the depthwise cross-correlation operation as proposed by Li et al. [2019a]), so that its output has the same number of channels as the feature maps. The output is then processed by a CNN with 5 convolutional layers: a 1×1 convolution which reduces the number of channels to 128 (as most backbone networks will typically have outputs with more channels), followed by four 3×3 convolutions with 128 channels as well. After each of these convolutions, batch normalization and a ReLu activation function are applied. All the convolutions in this subnetwork have a stride of 1 and apply padding so that their inputs and outputs have the same size. Finally, two output branches, each with a single 7×7 convolutional layer with a single output channel, are used to compute the two outputs of the network. One of these two branches performs classification and uses a sigmoid activation function after the last convolution. Its output should ideally be 1 for the regions of the search image where an object similar to the given template is located, and 0 everywhere else. In practice, these values will be between 0 and 1 and correspond to the detection confidence of the network. The other branch performs regression to compute a "centerness" score, which helps to locate the center of the detected objects more accurately. Unlike in the original SiamFC++ network proposed by Xu et al. [2020], here no additional branch is required for bounding box regression. As the objects to be detected will have a near identical size to the given template, simply locating their center is enough to compute a bounding box.

Unlike most typical object detection networks, this network architecture can take inputs of almost any size, as long as they fulfill a couple of conditions. Both the template and search images must be large enough so that they can be processed by the feature extraction subnetwork, which will reduce the spatial size of its inputs while increasing their number of channels. Additionally, the search image needs to be at least slightly larger than the template, so that the cross-correlation of the two feature maps is larger than the minimum required input size for the detection subnetwork. The overall stride of the network, and therefore of the template matching results (i.e., the step between the different sub-windows of the search image which are compared to the template)

will depend only on the design of the CNN used for feature extraction (e.g., the stride of its convolutional layers or other operations like max pooling). The stride should be significantly smaller than the size of the objects to be detected, Also, a larger stride will make the location of the detected objects less accurate.

Regarding the number of channels of the inputs, the network can take inputs of any number of channels as long as the first layer of the Siamese CNN is modified accordingly. For the case of single-pol VHR SAR images, both inputs will have a single channel, but the same method could also be applied images with multiple channels (e.g., polarimetric SAR images). The number of channels of the inputs may limit the use of pre-trained networks for the feature extraction, as most CNNs used in computer vision are trained using RGB image data and therefore expect inputs with 3 channels. If training a deep CNN for feature extraction from scratch is undesired or not possible, the input SAR data with a single channel can be duplicated to generate an input with 3 identical channels, which can then be fed into a CNN pre-trained using a large computer vision dataset (e.g., ImageNet [Deng et al., 2009]).

6.3 Training strategy

In addition to the aforementioned modifications to the network architecture, a new training strategy is also required to adapt a fully convolutional Siamese network for the task of object detection. For this, it is fundamental to understand the difference between the tasks of visual tracking and object detection.

In the original task of visual tracking, after the network has been trained, an arbitrary object (i.e., of an unknown class which may have not been present in the training data) is selected in the first frame of a video and used as a template to locate the object in the following frames. Typically, a single object is tracked (i.e., a single instance of a single class), this object is known to be present in the video, and its position is approximately known, as very large displacements between consecutive video frames are unlikely. Therefore, for tracking an object, the search image will be a small part of the video frame surrounding the object's previous position. Also, the tracking will work as long as the region of the search image with the maximum similarity corresponds to the new object's position.

In contrast, when performing SAR object detection by applying template matching, the task is to locate many objects (i.e., multiple instances of different classes) in SAR images, given a template database with a set of representative image chips of these objects. These SAR images can be of different locations and may or may not contain any of these objects. Unlike tracking, where only the maximum of the similarity map is relevant, for this task the similarity map needs to be thresholded to get an unknown number of detections, making false positives a more relevant issue. Additionally, these SAR images will typically depict very large scenes (e.g., a few kilometers), whereas the objects to be detected are much smaller (e.g., multiple meters). This results on a heavily imbalanced problem, as the majority of these large scenes will typically correspond to negatives. This is exacerbated by the potentially large number of classes, as objects of the same type but with different orientations are also considered as negatives.

Because of these differences, the training data needs to be sampled in a different way. Additionally, some SAR specific pre-processing need to be applied to the input SAR images before feeding the training data to the network.

6.3.1 Data pre-processing

Initially, a speckle filter should be applied to all the SLC SAR images to reduce the effect of the speckle noise, as otherwise it will very likely affect the detection performance. As the detection will be performed using only the SAR amplitude, the phase information can then be discarded.

VHR SAR images typically have a very large dynamic range (e.g., up to 90 dB for TerraSAR-X high resolution Spotlight images), which can be difficult to handle for standard CNNs [Zhu et al., 2021]. To deal with this, the linear amplitude can be converted into a logarithmic scale (i.e., in dB units). Additionally, the dynamic range of the resulting image can be further limited and normalized so that all pixels have values between 0 and 1 by applying the following transformation:

$$A' = \begin{cases} 0 & \text{if } A \leq A_{min} \\ \frac{A - A_{min}}{A_{max} - A_{min}} & \text{if } A_{min} < A < A_{max} \\ 1 & \text{if } A \geq A_{max} \end{cases} \tag{6.3}$$

where A' is the final normalized image, A denotes the SAR amplitude in dB units, and A_{min} and A_{max} are the minimum and maximum dB values used to limit the dynamic range. Because modern SAR sensors are well calibrated, the same dynamic range should work well for all the images from a given sensor.

If the original SLC SAR image has very different pixel spacings along the range and azimuth axes (which is the case for some SAR sensors), the image can be resized so that both axes have an equal pixel spacing, as the size of the receptive field of most CNNs is equal along both image axes. To avoid a resolution loss, the smallest of the two pixel spacings should be used. Additionally, it is important to ensure that the range axis is always the same and has the same direction (e.g., starting at near-range), so that the directions of the layover and shadow effects are consistent across all images. If this is not the case, some of the input SAR images might need to be rotated or flipped.

6.3.2 Sampling the training data

To train the proposed network architecture, the dataset should ideally consist of the full SAR images and not image patches. The annotation for each object should contain a reference to the corresponding SAR image, the bounding box defining its location in this image, the object class and its orientation with respect to the sensor. As previously mentioned, the orientation could also be obtained from a rotated bounding box. With this information, the image patch for any given object can be quickly read from the corresponding SAR image. Ideally, the aforementioned pre-processing should have already been applied to all the full SAR images in the dataset, so that when these patches are read no additional processing is required.

During training, the network will take batches of input pairs (each pair containing a template and a search image). Even though each template will be matched across a complete SAR image when performing object detection, during training these search images should be significantly smaller. Otherwise, a batch of such large images will not fit in the memory of the graphics processing unit (GPU) used to train the network. Besides, even if the GPU memory was large enough, using the full scenes as search images would result in a very imbalanced problem with an overwhelming majority of negatives, as there will typically be very few positive matches in these very large images. This imbalance can be reduced by controlling the sampling of the templates and the corresponding search images, so that a minimum percentage of positive matches can be ensured. The easiest way to ensure a higher amount of positive matches is to make certain that each of these smaller search images contains at least one object that is similar to the corresponding

template. However, this way the network will only learn to distinguish an object from its typical surroundings, which may result in false positives at other locations. Therefore, ideally both positive pairs (where the search image contains at least one similar object) and negative pairs (without any similar object) should be sampled in a balanced way. Before this can be implemented, negative samples (i.e., not containing any of the objects in the dataset) need to be generated.

Generation of negative samples

If all the objects of the relevant classes have been annotated for a given SAR image, negative samples can be automatically generated by randomly sampling image patches and checking that these do not overlap with any of the annotated objects. This process can be repeated for each of the SAR images in the dataset to generate a very large amount of negative samples. However, for certain scenes, this can result in a large amount of negative samples that can be very easily distinguished from the man-made objects to be detected. Examples of such easy negatives are those containing homogeneous clutter areas like vegetation, water, etc. To avoid having a large percentage of such easy negatives, a more sophisticated negative generation strategy is required.

In this work, two indicators will be computed to try to determine whether a certain image patch potentially contains man-made objects making it a difficult negative, or whether it corresponds to a homogeneous clutter patch. One of these indicators is the amount of CSs in a given patch. This can be efficiently computed for all the patches in a given SAR image by detecting the CSs for the whole image and using a boxcar filter. The other is the coefficient of variation (i.e., the standard deviation divided by the mean) of the SAR amplitude of all the pixels in a given patch. This can also be computed for all the patches across the whole image combining boxcar filters and a few simple operations.

Once these two indicators have been computed, different thresholds can be applied to the two corresponding images to identify different types of image patches. Patches with a high number of CSs are very likely to contain man-made objects. Patches with a lower number of CSs but a high coefficient of variation will likely exhibit different kinds of surfaces. Finally, patches with a low number of CSs and a low coefficient of variation likely correspond to homogeneous clutter areas. Suitable values for these thresholds can be chosen empirically by sampling some negatives of the different classes and visualizing them. This information can be used to selectively sample different types of negative samples (e.g., keeping a specific percentage of each type). When generating negatives in this way, the sampled patches should not overlap with any of the annotated objects. Additionally, it is recommended to avoid repeatedly sampling the same locations for a given image (e.g., by checking that the sampled negatives do not overlap). The negative generation only needs to be performed once for each of the SAR images in the dataset, as the bounding boxes for the generated negative samples can be saved together with all the other annotations in the dataset. During training, negative samples can then be read on the fly from the full SAR images as required using these bounding boxes.

Balanced sampling

During training, image pairs need to be read from the training dataset and fed to the network. These pairs will be sampled randomly, but with certain probabilities to try to compensate different types of imbalances in the training data. Each sampled pair will consist of two annotations: a template containing a given object, and a search image where this object should be detected. Before each training epoch, a large number of pairs can be sampled, and the corresponding image patches can then be quickly read as needed in the training loop using their annotations.

To sample a pair from the training dataset, the template will be first sampled by randomly selecting one of the annotated objects. Here, the previously generated negative samples should be excluded, as they do not contain relevant objects. When sampling a template, any class imbalance present in the dataset (e.g., objects of certain classes being much more common) can be addressed. For a dataset with N samples, n different object classes, and N_i samples of each class (with $i = 1, \ldots, n$), the probability of a random sample belonging to class i would be $p_i = {}^{N_i}/_{N}$. A perfectly balanced sampling would require all classes being sampled with an equal probability $p_b = {}^{1}/_{n}$. Such a balanced sampling can be achieved by modifying the probability with which each individual sample is taken: samples of class i need to be taken with a probability of p_b/N_i, instead of the original probability of $1/N$ for all samples. Even if this can be easily implemented, it can lead to a high repetition of the samples in the less common classes, which is far from ideal. Instead, a more flexible solution will be used here, where instead of p_b, each class is sampled with a probability p'_i, which can be computed as follows:

$$p'_i = (1 - a)\, p_i + a\, p_b \tag{6.4}$$

Here, a takes a value between 0 and 1, which can be used to control the sampling. A value of 0 would result in the original probabilities ($p'_i = p_i$), and a value of 1 on perfectly balanced sampling ($p'_i = p_b$). Intermediate values will progressively increase/decrease the probability of the less/more frequent classes. If desired, a similar approach can also be applied to balance the probabilities of the different imaging geometries within each object class. For this, both the incidence angle and the object orientation can be split into discrete intervals (i.e., bins), and the probability of each imaging geometry can be computed using the number of samples in each 2-D bin.

Once a template has been selected, a search image needs to be sampled as well to form an input pair. This can be done by taking another of the annotated samples in the dataset and increasing the size of its bounding box to the required size while keeping the same center. The amount of positive and negative pairs to be used during training can be controlled by generating positive pairs with a chosen probability p_p, and negative pairs with the complementary probability $p_n = 1 - p_p$.

A positive pair can be generated by randomly selecting one of the samples that share the same class and have a similar imaging geometry as the template. The use of the same sample for the template and the search image should ideally be avoided. The list of samples with a similar imaging geometry (i.e., those fulfilling the constraints in equations 6.1 and 6.2) can be computed once and stored for each of the samples in the dataset, so that this information can be quickly accessed.

Three types of negative pairs can be generated depending on the difference between the template and the sample used for the search image. The previously generated background samples can be used to learn to discriminate objects from different backgrounds. Samples of a different object class allow to learn how to distinguish different object classes. Finally, samples of the same class but with a different imaging geometry can be used to learn how to discriminate objects with different orientations. The probabilities with which these three types of negative pairs are generated can also be chosen.

In some cases, the generated negative pairs might actually contain objects which are similar to the corresponding template. This can occur when a similar object is located very close to the sample chosen to form the negative pair, as the search image is generated by increasing the size of this sample's bounding box. In the same way, the positive pairs might also contain several instances of a similar object and not just one. This will not have a significant effect on the balance between positives and negatives, but it needs to be taken into account when generating the ground truth for the sampled pairs, as it will be explained later.

Finally, to train the network using gradient descent, the sampled pairs need to be organized into batches (i.e., groups of samples which are passed through the network together to update its parameters). All the samples in a batch must have the same size. This is not a problem for the search images, as their size can be arbitrarily chosen. However, the templates for different objects will most likely have different sizes. To avoid resizing them or having too much background surrounding the smallest objects, the size of their bounding boxes can be rounded up to a multiple of a certain number of pixels (e.g., 16 or 32). At the beginning of each training epoch, all the templates for that epoch can be sampled, and they can all be organized into batches by grouping together those with the same sizes. The search images can then be sampled for each of these batches to form the pairs of batches to be used during training.

Data augmentation

Many of the transformations typically used for data augmentation cannot be applied to SAR images, as they would result in unrealistic SAR images which could never be generated by a real sensor. This is the case of random image rotations, which are commonly applied to optical imagery acquired close to nadir, but would result in SAR images with the layover and shadow effects occurring in wrong directions. Therefore, in this work few data augmentation techniques will be applied. Besides, many different input pairs can be generated even for small datasets, making it unlikely that the network sees the exact pair of inputs many times even without data augmentation.

However, depending one the chosen architecture for the Siamese CNN, random translations might need to be applied to the search images. As previously mentioned when describing the network architecture, using deep CNNs with padding for the feature extraction results in a spatial bias, as the network is no longer invariant to translation. This can be addressed by applying a random translation to the input data, to make sure that the object which is similar to the template is not always in the center of the search image [Li et al., 2019a]. Therefore, when sampling the search image of a positive pair, a random translation will be applied the corresponding bounding box before reading the corresponding patch from the full SAR image. The maximum possible translation should be at least a certain percentage of the size of the search image to completely eliminate this spatial bias [Li et al., 2019a]. However, it should also be small enough to ensure that the similar object is still completely within the bounds of the search image after this translation is applied.

Optionally, the SAR images can also be flipped along the azimuth axis. This transformation should be valid, as it will result in realistic SAR images. For objects which are symmetrical around a plane perpendicular to the ground, flipping their SAR signature along the azimuth axis should be nearly equivalent to imaging them with a different orientation with respect to the SAR sensor. This assumption should be reasonably accurate for many objects (e.g., airplanes, ships, trucks), which can be considered symmetrical under the limited resolution of spaceborne sensors. This can be used to generate samples for some missing imaging geometries. If this transformation is applied, this different orientation must be taken into account when applying equation 6.2 to determine whether two samples are similar or not.

6.3.3 Ground truth generation and loss functions

When sampling pairs for training the network, the corresponding ground truth has to be generated on the fly. Given a template and a search image, this requires determining which regions of the search image are similar to the template in an efficient way. Before starting the training process, all the similar pairs of objects in the dataset can be pre-computed and organized in a suitable

data structure, so that for any template, all the similar objects in a given SAR image can be quickly obtained. Using this, the ground truth for the two outputs can be generated for each input pair.

First, the ground truth for the similarity map (i.e., the output of the classification branch) needs to be generated. Each pixel in this output image represents the similarity between two image patches: the template and a specific sub-window of the search image. The ground truth will take a value of 1 for all the pixels whose sub-windows contain objects similar to the template, and 0 for the rest. Sub-windows partially containing a similar object (i.e., where the object is not centered) are also considered similar as long as they significantly overlap with the object. If a similar object is centered at pixel (x_c, y_c) of the search image, all the output pixels (x_o, y_o) fulfilling the following two conditions will be set to 1:

$$\left| x_c - \left(x_o s + \left\lfloor \frac{s+w}{2} \right\rfloor \right) \right| \leq kw \tag{6.5}$$

$$\left| y_c - \left(y_o s + \left\lfloor \frac{s+h}{2} \right\rfloor \right) \right| \leq kh \tag{6.6}$$

This is equivalent to finding the sub-windows whose center is within a distance of the object's center. This distance is proportional to the object's size to ensure a certain amount of overlap between the object and the sub-windows. Here, s denotes the network stride, w and h are the object's width and height (in pixels). The hyperparameter k takes a value between 0 and 1 and controls the minimum required overlap.

After generating the ground truth for the similarity map, it also needs to be generated for the "centerness" score. This score will be 0 for all the pixels which were set to 0 in the similarity map, and will need to be computed for those that were set to 1. This score should take higher values for the output pixels whose associated sub-windows have a higher overlap with the corresponding object (i.e., those that are closer to its center). The "centerness" score for a given pixel (denoted here as c) can be computed using the following equation:

$$c = \sqrt{\frac{\min(l,r)}{\max(l,r)} \frac{\min(t,b)}{\max(t,b)}} \tag{6.7}$$

where l, r, t and b represent the distances between the center of the corresponding sub-window and the left, right, top and bottom of the object's bounding box, respectively. That way, pixels corresponding to a sub-window which is perfectly aligned with an object will get a "centerness" score of 1, and this score will progressively drop for sub-windows which are further away and have a lower overlap.

The outputs of the two network branches will be compared with the generated ground truths using two different loss functions. Focal loss [Lin et al., 2017] is used to compute the loss of the classification branch (denoted as L_{cls}), as this output corresponds an imbalanced classification problem with much more negatives than positives. The loss of the regression branch predicting the "centerness" score (denoted as L_{ctr}) is computed using the binary cross entropy loss, as proposed by Xu et al. [2020]. Both losses are computed for each of the pixels in the corresponding images and then summed across all the pixels, resulting in a loss value for each output branch. The total loss L is computed as a weighted sum of the two losses:

$$L = L_{cls} + \lambda L_{ctr} \tag{6.8}$$

with the weight λ being an additional hyperparameter.

6.4 Inference with the trained network

In order to perform inference with the trained network, a template database needs to be first generated. Then, this template database and the trained network can be used to detect objects in new SAR images.

6.4.1 Building a template database

The training dataset contains many samples of the different objects to be detected. These samples can now be used to detect the corresponding objects in new SAR images. However, the training dataset will most likely contain a significant amount of samples, many of which will be redundant (e.g., many instances of the same objects with the same imaging geometry). Because of this, using all the samples in this dataset as templates will be extremely inefficient or even unfeasible. Instead, a template database with just enough representative samples of the different objects to be detected can be built by selecting a subset of the training dataset.

To create this database, the training samples can be first separated according to the different object classes. Then, for each class, the samples with similar imaging geometries (i.e., incidence and object orientation angles) should be grouped together. This can be done by dividing the range of possible incidence angles (e.g., 15° to 60°) as well as the possible object orientations (0° to 360°) into multiple discrete intervals. Each sample can then be assigned into the corresponding bin, as if computing a 2-D histogram. The size of these bins should be equal or smaller than the values of the δ_θ and δ_α hyperparameters selected when training the network. Smaller bins will result in a larger template database (due to a finer sampling of the different imaging geometries for each object), which will then make the object detection slower.

A simple approach to create this database is to randomly select one sample from each of these 2-D bins for each class. In this work, this simple approach will be used. However, more advanced strategies could also be implemented, involving the selection of multiple samples from some of the bins. In some cases, two samples of the same object with the same imaging geometry can still be significantly different (e.g., due to seasonal effects, as illustrated in Fig. 6.4). Even though the network is trained to identify these as similar objects, it could still be beneficial to use several templates for a single imaging geometry when these vary significantly. The trained network can be used to evaluate the similarity of the templates in each bin, so that different samples inside a single bin can be automatically identified and added to the template database.

Finally, if new samples with different imaging geometries and/or different object classes become available, an existing template database can be updated without necessarily training the network again.

6.4.2 Detecting objects in a SAR image

Given a SAR image and a template database, the network should be used to detect all the objects in the image which are similar to these templates. This will result in a set of predicted bounding boxes, each with an associated object class and a confidence value. The whole process can be divided in the four steps described below.

Pre-processing

Initially, the same pre-processing (e.g., speckle filter, dynamic range, resizing, etc.) that was applied to the training data should be applied to the input SAR image. Then, the templates that should be searched in this image need to be selected from the template database. If the SAR image has an incidence angle θ (which is typically given in the image metadata), only the

templates with similar incidence angles should be selected. These are those with an incidence angle θ_{temp} fulfilling the following condition:

$$\theta_{temp} \in [\theta - \delta_\theta, \theta + \delta_\theta] \tag{6.9}$$

where δ_θ is the maximum difference in incidence angle so that two samples of the same object were considered as similar during training. This selection can be further refined if prior knowledge about the imaged scene is available (e.g., which types of objects might be present in the imaged location).

The feature map should be computed for each of these templates by passing them through the corresponding branch of the feature extraction subnetwork. This has to be done separately for each template, as the they might have different sizes, and therefore cannot be processed as a batch. Alternatively, the feature maps for each template could also be pre-computed and stored in the template database, so that they do not need to be computed once again for each new image.

Tiling

Afterwards, the input SAR image needs to be split into smaller tiles. Even though the chosen network architecture can potentially take a very large search image as an input, this image still needs to fit into the memory of the GPU used to run the computations. Therefore, to detect objects across a full VHR SAR image (which may have upwards of 100 million pixels), this image will typically need to be split into a set of overlapping tiles of a more manageable size (e.g., 4096×4096 pixels). The overlap between consecutive tiles should be larger than the size of the largest object to be detected, to ensure that each object is completely contained within a single tile. The network can then be used to detect objects in each of these tiles separately, resulting in a set of predicted bounding boxes for each tile with their corresponding classes and confidence values. The predictions for all these overlapping tiles can then be combined later in a post-processing step to get the results for the whole image.

Detecting objects in an image tile

To get the detections for a given tile, first its feature map needs to be computed by passing it through the corresponding branch of the feature extraction subnetwork. This only should be done once. The resulting feature map should be cross-correlated with the features maps from each of the templates (which are already available), and the results should be passed through the detection and classification subnetwork. Again, this operation has to be done separately for each template, as they might have different sizes. To reduce the memory consumption, the detections can be computed from the network's output before processing the next template, so that only a small number of bounding boxes have to be stored for each template.

For a given template the network will output two images, and some additional processing needs to be applied to derive the detections from them. The main output is the one of the classification branch: an image with a pixelwise similarity score, which will be treated here as the detection confidence. To get the detections, a threshold should be applied to this similarity score. For each of the pixels with a similarity score above the threshold, a bounding box can be easily computed. The pixel coordinates (x, y) of the top-left corner of each bounding box can be computed as follows:

$$(x, y) = \left(\left\lfloor \frac{s}{2} \right\rfloor + x_o s, \left\lfloor \frac{s}{2} \right\rfloor + y_o s \right) \tag{6.10}$$

where s is the overall stride of the network, and x_o and y_o are the corresponding pixel coordinates in the output similarity map. The width and height of the bounding boxes are all equal to those of the input template.

Because each detection will typically involve high similarity scores for many neighboring pixels, non-maximum suppression (NMS) [Neubeck & van Gool, 2006] needs to be applied to eliminate redundant detections. This requires choosing a threshold for the intersection over union (IoU), which controls the maximum possible overlap between two detections before they are considered redundant. The output of the regression branch can be used to select the bounding box which more accurately fits the object's location. NMS takes all the bounding boxes and a weighted sum of their corresponding similarity and "centerness" scores as inputs, and selects the best bounding boxes. After applying NMS, the "centerness" score can be discarded, and the similarity score for each bounding box is taken as the detection confidence.

After the detections for all the templates in a given tile are computed, NMS needs to be applied once again to filter out wrong detections, as the multiple templates can lead to highly overlapping detections. Here, NMS will remove the detections with the lowest confidence (i.e., similarity score), using the same IoU threshold as before.

Post-processing: combining the detections from all the tiles

Once the detections have been computed for all the overlapping tiles, these need to be combined to get the results for the whole image. For this, an offset should be added to the pixel coordinates of the bounding boxes in each tile, according to the tile position in the full image. To eliminate repeated detections due to the tile overlap, NMS should be applied one final time.

If prior knowledge of the imaged scene is available (e.g., from map data), an additional post-processing step can be implemented to check the plausibility of each detection. Certain object classes are only present in specific locations (e.g., ships on water, airplanes inside airports), and therefore detections outside of these regions are most likely incorrect and can be removed.

As a final note, for each of the resulting detections, in addition to the bounding box and a confidence score, all the information associated with the corresponding template will also be available. This includes not only the object class, but also its approximate orientation. Therefore, the approximate orientation of the detected objects is also predicted. The precision with which the orientation is estimated will depend on the similarity criterion used during training and on the sampling of the different orientations in the template database.

7 Experiments

This Chapter shows the experiments performed to test the methods introduced in Chapters 4, 5 and 6, as well as the SAR data used in these experiments. Section 7.1 contains the experiments for the monitoring of oil storage tanks, Section 7.2 those for the change detection method, and Section 7.3 those for object recognition. In each of these sections, the data used to evaluate the corresponding method is first introduced. Then, the experiments performed to test the different parts of each method and to select suitable values for its parameters are shown. Finally, the main experiments are introduced, describing how these methods were applied to solve specific problems involving the monitoring of different types of human activity. The results of these final experiments will be shown in detail in Chapter 8. All the SAR images shown in this chapter were transformed for visualization as described in Section 3.1. Part of the material in this Chapter has been published in [Villamil Lopez & Stilla, 2021] and [Villamil Lopez & Stilla, 2022].

7.1 Monitoring of oil storage tanks using coherent scatterers

7.1.1 Dataset

To evaluate the performance of the proposed method for the monitoring of oil storage tanks, a dataset consisting of three TerraSAR-X repeat-pass images of the port of Rotterdam will be used. These three images have been acquired on July 23, August 3 and August 14, 2017; using the Staring Spotlight imaging mode (with a resolution of 58 cm in slant-range and 23 cm in azimuth) and with an incidence angle of 48.1°. An overview of the imaged scene can be seen in Fig. 7.1. The imaged scene has an extent of approximately 3×5 km and contains 167 oil storage tanks of different sizes. Among these, there are 96 tanks with floating roofs, and 71 with fixed roofs.

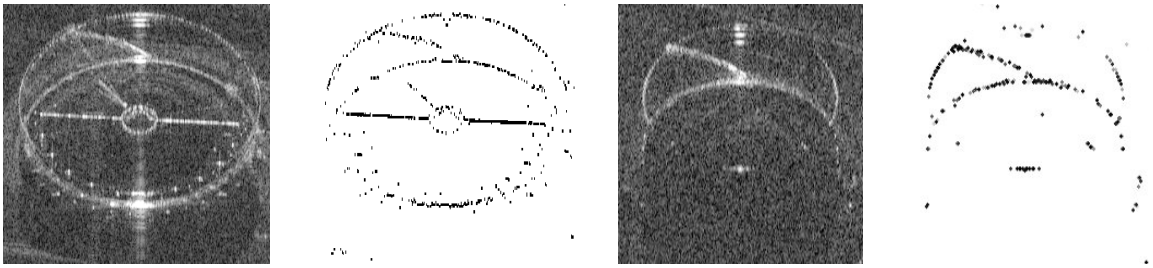
7.1.2 Examples and parameter selection

Here, this dataset is used to illustrate different parts of the proposed method and to select suitable values for all its parameters.

Detection of coherent scatterers

For the detection of CS, the following parameters were used: a threshold $T = 0.5$, and $n = 40$ sublooks covering the whole available bandwidth with a 75% spectral overlap between consecutive sublooks. For the TerraSAR-X imagery in this dataset with a bandwidth of 300 MHz, this resulted on a sublook bandwidth of 27.90 MHz and a separation of $f_s = 6.97$ MHz between sublooks. Figure 7.2 shows an example of the detected CSs for a storage tank with a floating roof and one with a fixed roof. The selected parameters result in a good detection performance. Most of the point scatterers visible in the SAR image are correctly detected, and almost no CSs are detected in the clutter areas. Moreover, the example shows that the image with the detected CSs has virtually the same resolution as the input SLC image.

Figure 7.1: Overview of the "port of Rotterdam" scene used to test the monitoring of oil storage.



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

Figure 7.2: Results of the CS detection for two different oil storage tanks. Tank with a floating roof: a) SAR amplitude, b) detected CSs. Tank with a fixed roof: c) SAR amplitude, d) detected CSs.

Approximate location of the oil tanks

As explained in Section 4.2.1, the initial detection of storage tanks in the imaged scene is out of the scope of this thesis, as several works dealing with this topic have already been published. The approximate locations of the oil tanks in this dataset were obtained automatically from OSM by performing a query using its Overpass API*. For a given storage tank, the latitude and longitude coordinates of its center (obtained from OSM) can be projected into the SAR image by performing geocoding with a DEM, obtaining the approximate pixel position for the center of the tank bottom $\hat{p}_b = (\hat{x}_b, \hat{y}_b)$. For the performed experiments, it was assumed that these locations have an uncertainty $u = 15$ m. As it will be shown below, the OSM data for this location is actually more accurate, but this value was intentionally set to a higher value to demonstrate that the proposed method only needs an approximate location.

In addition to these approximate locations, the approximate radius of each tank $\hat{r}_t$ can also be extracted from the OSM data. This information is not needed by the proposed method but can be used if available. For this, the polygon associated to each tank is first projected into a Universal transverse mercator (UTM) map projection, in order to avoid errors in the estimated radius due to the map projection used by OSM. The bounding box enclosing each of these polygons is then computed, and the average of its height and width is taken as the estimation for diameter of the corresponding tank.

Figure 7.3 shows an example of the OSM data available for this dataset, and illustrates how this data is used. A screenshot of the actual OSM map for a small region of the imaged scene be seen in Fig. 7.3a. Figure 7.3b shows the polygons for the oil storage tanks projected into the SAR image. Here, it can be seen that at least for this particular location, the information available in OSM is quite accurate. However, this accurate information will not be used in this thesis, as the goal is to demonstrate the proposed method. Instead, the approximate center locations obtained from OSM are used to compute rough bounding boxes for the oil tanks, as it will be shown below.

Estimation of oil tank dimensions using a single image

Once the approximate location of the storage tanks has been obtained and the CSs have been detected, the relevant dimensions of each storage tank can be estimated by applying the method described in 4.2.2. This method has four parameters, but these are all related to the minimum and maximum possible sizes of the storage tanks and therefore their values do not need to be set empirically. In this thesis, the following limits for the height h_t and radius r_t have been chosen, which covered the sizes of all the storage tanks present in the used dataset: $h_t^{min} = 12.5$ m, $h_t^{max} = 25$ m, $r_t^{min} = 10$ m and $r_t^{max} = 50$ m. Tighter limits for r_t can also be set individually

*`https://wiki.openstreetmap.org/wiki/Overpass_API` (last accessed on February 5th, 2023).

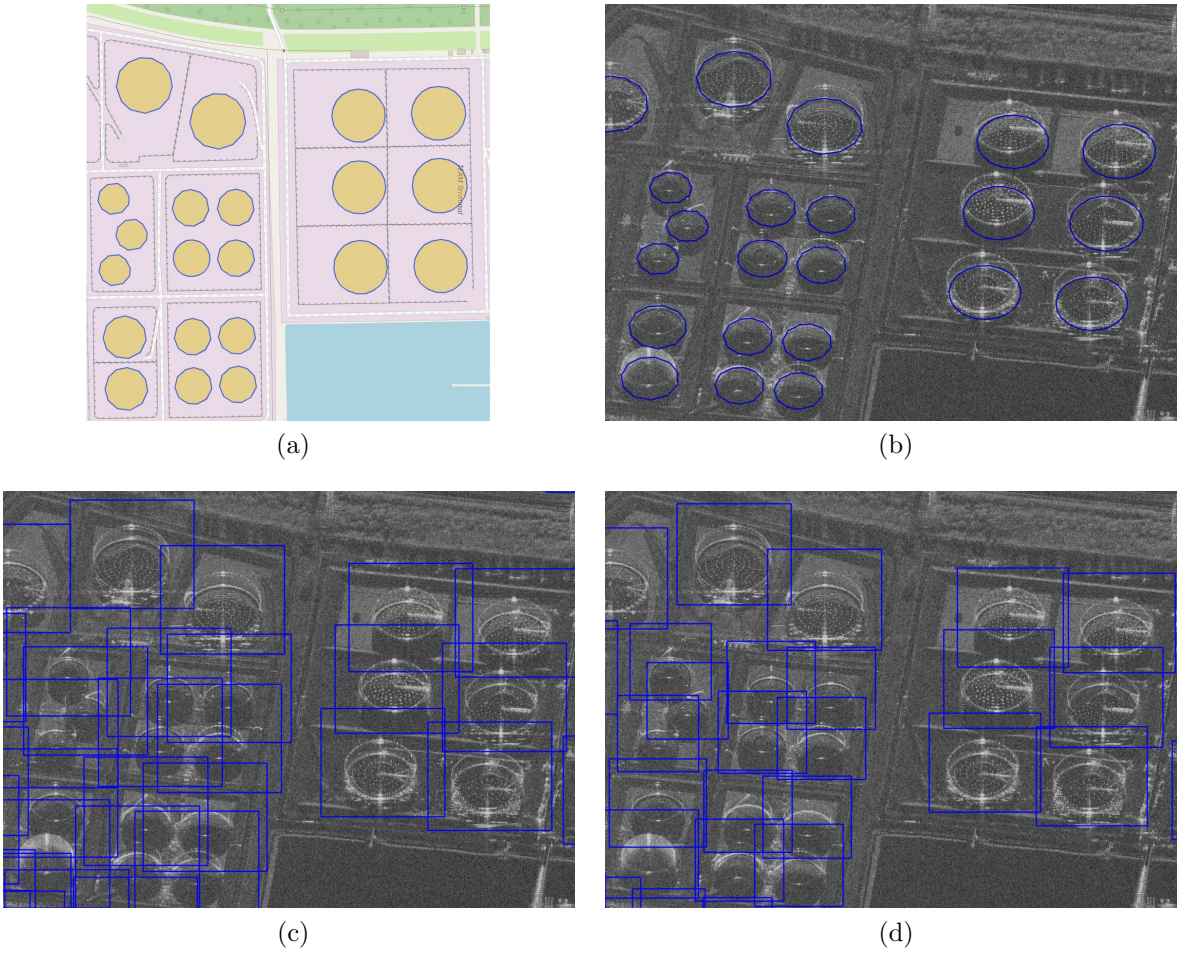


Figure 7.3: Obtaining the approximate location and size of oil storage tanks from OpenStreetMap (OSM). a) Screenshot of the OSM map. b) OSM polygons projected to the SAR image. Bounding boxes used to test the proposed method, computed with prior knowledge of: c) each tank's approximate location, d) each tank's approximate location and radius.

for each tank by using the approximate radius $\hat{r}_t$ obtained from OSM (or whichever method was used to get the approximate locations of the storage tanks). Later in Section 8.1, the results and runtimes will be compared for the cases in which the generic limits for r_t are used, with those obtained when an approximate radius $\hat{r}_t$ is available for each tank.

Using the chosen limits for h_t and r_t and the information obtained from OSM, equations 4.7 and 4.8 can be applied to compute approximate bounding boxes for all the storage tanks. For the previous example, the bounding boxes computed using the generic limits for h_t and r_t can be seen in Fig. 7.3c. For comparison, those obtained using $\hat{r}_t$ to set specific limits for each tank ($r_t^{max} = \hat{r}_t + 5$ m) are shown in Fig. 7.3d. This illustrates how prior knowledge about the size of the tanks is clearly beneficial, as it results in tighter and more accurate bounding boxes.

With this, the semicircular double reflections at the bottom and at the top of a storage tank can be detected. Figure 7.4 illustrates this process with an example for a storage tank with a floating roof. The CSs detected inside the corresponding image patch are shown in Fig. 7.4a. Figure 7.4b illustrates how the computation of the Hough accumulator works, showing the possible centers for three different CSs (each highlighted in a different color) and two different radii (one shown as continuous and the other as dashed lines). Finally, the two semicircles detected for this example using equation 4.12 with the computed Hough accumulator h_2 can be seen in Fig. 7.4c.

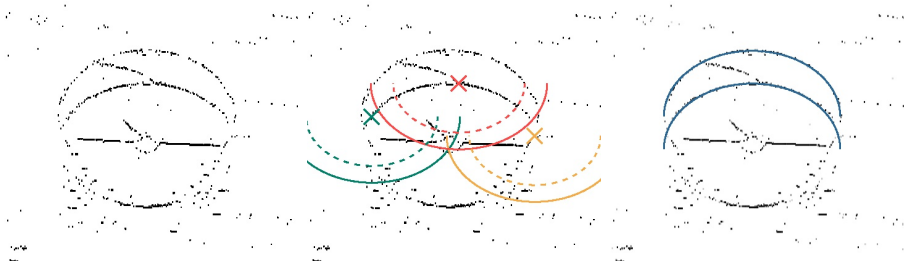


Figure 7.4: Detection of semicircular double reflections at the top and bottom of a storage tank. a) CSs inside the image patch surrounding a storage tank, b) possible center pixels illustrated for three different CSs and two different radii, c) and two semicircles detected using the Hough accumulator h_2.

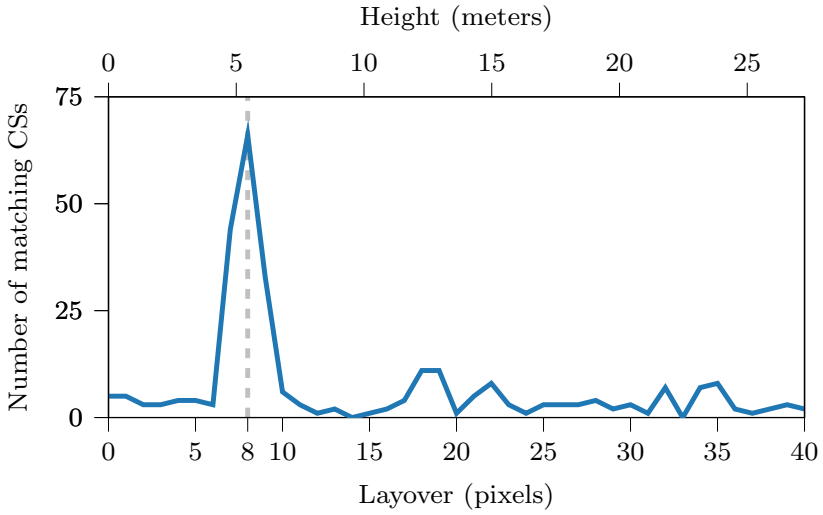


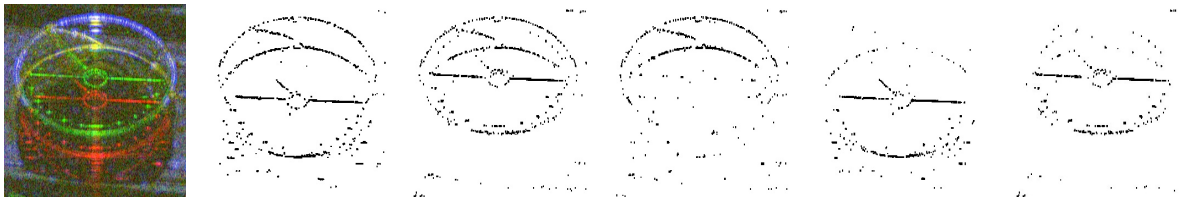
Figure 7.5: Estimation of the vertical position of the floating roof of an storage tank. Example for the storage tank previously shown in Fig. 7.4.

For this example, it can be clearly seen that both the position of the tank's center as well as its radius r_t and height h_t are correctly estimated.

After these two semicircles have been detected, the semicircle for the floating roof also needs to be detected by performing cross-correlation as described in 4.2.3. This requires an additional parameter, the maximum height for the floating roof h_r^{max}, which needs to be set slightly higher than the estimated tank height h_t. In this thesis, this limit will be set to $h_r^{max} = h_t + 5$. In Fig. 7.5, the results of this cross-correlation can be seen for the storage tank previously shown in Fig. 7.4. Here, the maximum can be clearly identified, and the estimated layover of the floating roof corresponds to the one that can be manually measured in the image.

Identification of the static and moving parts of the oil tanks

When using a time series for the monitoring of oil storage tanks, change detection can be applied as described in Section 4.3.1 to separate the static and moving parts of the oil tanks. This method has only two parameters related to the interferometric coherence: the kernel size used for its calculation and a threshold γ_t. Here, the coherence will be computed using a boxcar filter with a window of 7×7 pixels, and the coherence threshold will be set to $\gamma_t = 0.35$. Both values were empirically selected and performed well for the available data.



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

Figure 7.6: Separation of static and moving parts of an oil storage tank using change detection. a) Input image pair shown as a color composite image, b-c) detected CSs for each image, d) static CSs, e-f) moving CSs in each image.

Example results of this change detection algorithm can be seen in Fig. 7.6 for an image pair showing a storage tank with a floating roof at two different dates. Figure 7.6a shows this image pair as a multitemporal color composite image with both amplitude images in the red and green channels, and the interferometric coherence in the blue channel. The CSs detected in each individual image are shown in Fig. 7.6b and 7.6c. After the change detection, these CSs are classified into those which remained static (shown in Fig. 7.6d) and those that moved (shown for each image in Fig. 7.6e and 7.6f). As expected, it can be clearly seen that most of the static CSs correspond to the two double reflections at the outer tank structure, whereas most of the moving CSs correspond to the floating roof, which moved vertically due to a change in the amount of oil stored. It is important to note that for time series covering short time periods, the floating roof of some storage tanks might not move, as the amount of oil stored in them might remain constant during this time. In these cases, it will not be possible to separate the floating roof from the outer tank structure, as both will remain static. Because of this, the use of time series covering a long time period is advantageous.

Estimation of the vertical displacement of the floating roof between two images

Applying change detection with a time series allows to separate the CSs corresponding to the floating roof. As a consequence, cross-correlation can be used to accurately estimate the vertical displacement of the floating roof between each of the image pairs in the series. The results obtained when applying this cross-correlation to the image pair from Fig. 7.6 can be seen in Fig. 7.7. The maximum of the cross-correlation can be clearly identified, and as expected the estimated vertical displacement corresponds to the one which can be manually measured.

7.1.3 Practical application: monitoring of oil storage for the complete refinery

After suitable values were selected for all the method's parameters, this method was applied to the TerraSAR-X images of the port of Rotterdam to estimate all the relevant information for all the oil storage tanks in this scene. Before applying the proposed method to the complete scene, the GIS data obtained from OSM was cleaned by automatically removing those tanks which were either partially outside of the bounds of the imaged scene or too small to be considered of interest. In this case, the few tanks with a radius smaller than 10 m were considered too small and removed from the data, as their size is negligible compared to the many larger tanks with radii up to 50 m, and at those very small sizes the proposed method is more prone to making errors. Finally, in order to evaluate the performance of the proposed method, without the influence of errors in the OSM data, the OSM data was verified by visually comparing it to the SAR images in a GIS software, and a couple of storage tanks which were present in OSM but not in the used SAR

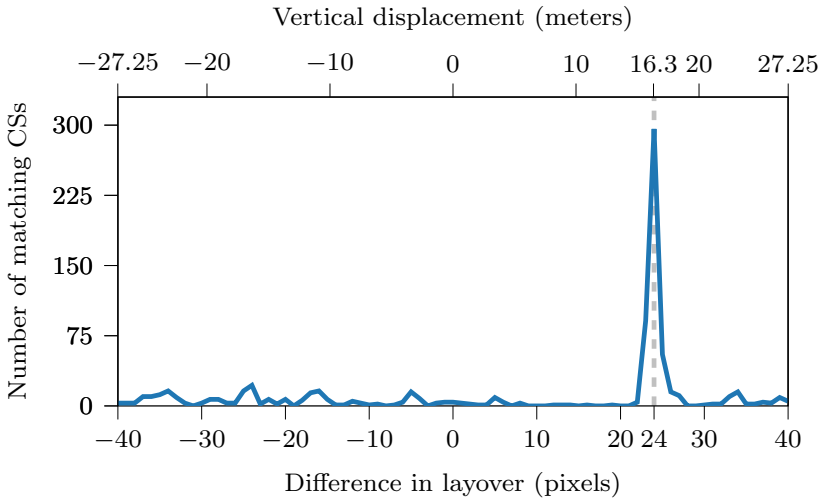


Figure 7.7: Estimation of the vertical displacement of the floating roof between an image pair. Results obtained using cross-correlation with the image pair from Fig. 7.6.

images were manually removed. After this, 167 oil storage tanks remained, including 96 tanks with floating roofs and 71 with fixed roofs.

The relevant parameters of each of these storage tanks were obtained by applying the two methods introduced in Sections 4.2 and 4.3, using the first image and the complete time series as inputs, respectively. Unfortunately, no ground truth data regarding the sizes of these storage tanks or the amount of oil stored in them was available. However, the same measurement principle applied here (using the semicircular double reflections as described in Section 4.1) has also been applied and validated by Hammer et al. [2017], who manually selected several image points on these double reflections and compared the dimensions derived from these points with those provided by the company operating the oil tanks. Therefore, if the semicircular double reflections are accurately detected by the proposed method, it will imply that the estimated dimensions should be accurate. The correct detection of these semicircular double reflections was qualitatively evaluated for all the 167 tanks by performing a visual accuracy assessment. Besides, the accuracy of the estimated dimensions was quantitatively evaluated for 10 storage tanks (5 with floating roofs and 5 with fixed roofs) by comparing the obtained values with those obtained in manual measurements.

To evaluate the performance of the proposed classification approach to distinguish between tanks with a fixed and a floating roof, a dataset had to be first generated by assigning the appropriate label to each of the storage tanks, as this information was not available in the OSM data. These labels were manually assigned by visualizing each storage tank in the SAR images, resulting in a dataset with 167 samples which will then be split into training and test data. The accuracy of the proposed classification approach was evaluated and compared for different classifiers and different amounts of training data.

All these results, and a short overview of the method's runtimes, will be shown in Section 8.1.

7.2 SAR change detection using coherent scatterers

7.2.1 Dataset

To evaluate the proposed CD method, a dataset consisting of 49 TerraSAR-X repeat-pass images of the city of Munich was used. These images have been acquired between March 28, 2016 and February 28, 2019; using the Staring Spotlight imaging mode (with a resolution of 58 cm in slant-range and 23 cm in azimuth) and with an incidence angle of 37.5° and in ascending orbit. An overview of the imaged scene can be seen in Fig. 7.8. This scene has an extent of approximately 3×5.8 km. Many different types of changes can be seen across the 49 images acquired during this period of almost three years, some of which will be illustrated in the following experiments. However, the main focus will be placed on the detection of changes due to construction activity, as many new buildings have been constructed and existing buildings renovated during this time.

7.2.2 Examples and parameter selection

Here, this dataset is used to illustrate different parts of the proposed method and to select suitable values for all its parameters.

Detection of coherent scatterers

For this dataset, ten equally spaced sublooks with a 75% spectral overlap appear to be a good choice for CS detection. As the used TerraSAR-X data has a total bandwidth of 300 MHz, this results in a sublook bandwidth of 92.30 MHz and a spacing of 23.08 MHz. For the threshold T, values between 0.1 and 0.125 work well for the chosen number of sublooks. The value of T trades off the number of false positives and false negatives. These parameters are different from those previously used for the monitoring of oil storage tanks. Using more sublooks with less bandwidth (as done previously) results in less CSs being detected in layover regions (e.g., building façades), as neighboring CSs interfere with each other in the sublook images due to their lower range resolution.

An example of the CSs detected in a TerraSAR-X image of a building can be seen in Fig. 7.9. The CS detected using the parameters proposed here (10 sublooks and $T = 0.125$) are shown in Fig. 7.9b. For comparison, the CS detected using the parameters previously proposed for the monitoring of oil storage (40 sublooks and $T = 0.5$) are shown in Fig. 7.9c. While both sets of parameters work, it can be clearly seen that using less sublooks results in a better performance in layover areas.

Coherence calculation and thresholding

The coherence plays an important role in the proposed CD method. The window size used for the coherence estimation and the threshold γ_t are important parameters. Here, the effect of both parameters will be analyzed, and suitable values will be selected. Figure 7.10 shows the coherence for an image pair computed using two different window sizes. This image pair has a temporal baseline of 22 days. The coherence map in Fig. 7.10a was computed using a smaller window of 3×7 pixels and is clearly noisier. The coherence map in Fig. 7.10b was computed using a window of 9×23 pixels and has a lower resolution, but also significantly less noise. In both cases, the window is bigger along azimuth, as the data has a higher resolution along this axis. The window sizes are chosen to achieve a similar resolution in slant-range and azimuth. Table 7.1 shows the mean and standard deviation of the coherence for clutter and point scatterers, computed using four different window sizes. The small homogeneous patch highlighted in yellow in Fig. 7.10 was used for estimating the clutter statistics, whereas for the point scatterers, 100 of them were

Figure 7.8: Overview of the “Munich” scene used to test the monitoring of construction activity.

(a) (b) (c)

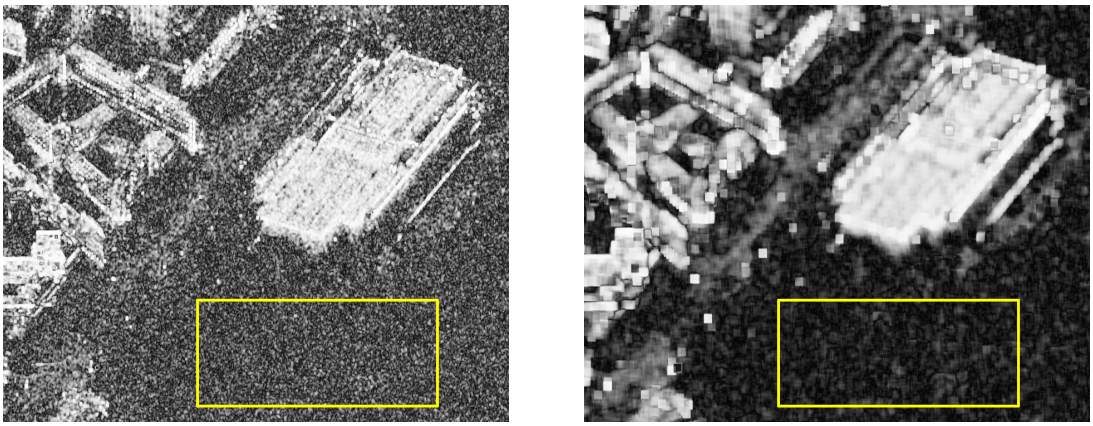
Figure 7.9: Example of coherent scatterers detection for an image patch showing a building. a) SAR amplitude, b) detected CS with 10 sublooks and $T = 0.125$, c) detected CS with 40 sublooks and $T = 0.5$.

manually selected across the image. Table 7.1 shows that for CSs, even relatively small windows result in good coherence estimates with low bias and variance. However, for clutter and other areas with low coherence, the coherence estimates using small windows have a high bias and variance. Because of this, using small window sizes can lead to problems when applying a threshold in areas of low coherence (e.g., coherence can be overestimated where changes occurred). Therefore, in this work a window of 9×23 is used for the coherence estimation. This results in coherence maps with a resolution of approximately 5.2 meters in azimuth and slant-range. Nevertheless, as described in Section 5.1, this does not affect the spatial resolution of the results of the pixelwise CD using CSs.

For the coherence threshold a value of $\gamma_t = 0.5$ is selected, taking advantage that CSs typically have high coherence and are not significantly affected by temporal decorrelation [Ferretti et al., 2001; Eineder et al., 2009]. This is illustrated in Fig. 7.11 with the histograms of the coherence values for the CSs (blue line) and for all the pixels (orange line) inside a very large image patch showing the city center. This comparison is done for two different temporal baselines: one of 22 days, shown Fig. 7.11a, and one of almost 3 years, shown in Fig. 7.11b. As expected, the histograms for the CSs show clear maxima for values very close to 1. This indicates that temporal decorrelation is not significant for CSs even after a period of almost 3 years. There are a few CSs with low coherence values, but these are most likely due to changes between the two images and also some false positives in the CS detection. The number of CSs with low coherence increases for the longer temporal baseline, as many more changes occurred during this time. When comparing the two histograms for all the image pixels, it is clear than the temporal decorrelation is in this case much more significant. Also, when considering all image pixels, the coherence values are much more evenly distributed even for short temporal baselines. Because of all this, a fixed coherence threshold like the one used in this work works well for CSs, but it is unlikely to work well when applied to all the image pixels. When considering all the image pixels, a different CD metric such as GLRT [Monti-Guarnieri et al., 2018] would likely perform better.

Change detection with an image pair using CSs

The CD method proposed in Chapter 5 should detect only those changes corresponding to man-made objects and not be affected by temporal decorrelation. To illustrate this, the method described in Section 5.1 was applied to an image pair acquired over the Munich area of the "Deutsches Museum". The first image was acquired on March 28, 2016 and the second on March 13, 2018. The two amplitude images can be seen in Fig. 7.12a and 7.12b, and the detected CSs for each image are shown in Fig. 7.12d and 7.12e, respectively. The CSs are represented as large points for better visualization, but each CS actually corresponds to an individual pixel in the

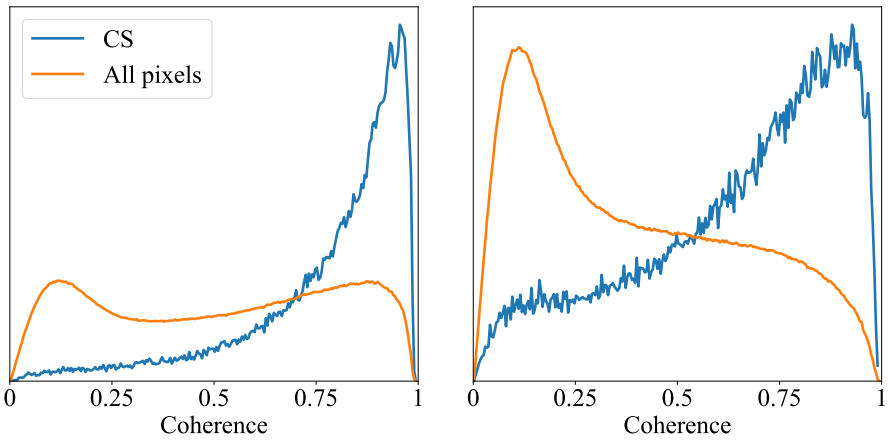


(a) (b)

Figure 7.10: Effect of the window size in the coherence calculation. Coherence of an image pair computed with a window of a) 3×7 pixels, and b) 9×23 pixels. The image patch marked in yellow was used for estimating the coherence statistics for clutter listed in Table 7.1.

Table 7.1: Coherence statistics for different window sizes.

Window size	Looks	Clutter		Point scatterers	
		Mean	Std. dev.	Mean	Std. dev.
3×7	21	0.313	0.145	0.935	0.070
5×11	55	0.213	0.105	0.906	0.087
7×17	119	0.155	0.080	0.876	0.093
9×23	207	0.124	0.065	0.861	0.102



(a) (b)

Figure 7.11: Comparison of the distribution of the coherence values for the CSs and all the image pixels. The histograms for the CSs (blue) and for all the pixels (orange) are shown for two temporal baselines: a) 22 days, and b) 2 years and 11 months. The histograms are computed using a large image patch in the city center and normalized so that the area under the curve integrates to 1.

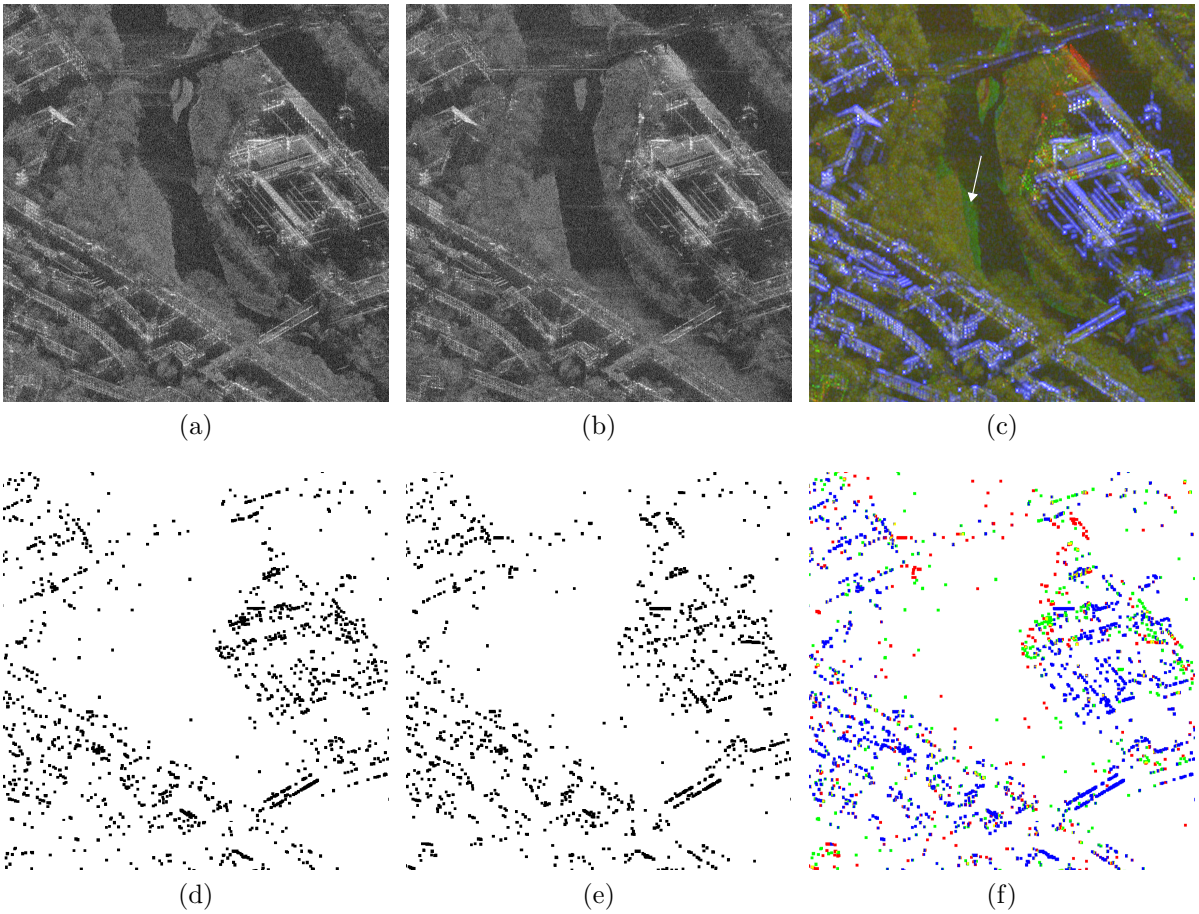


Figure 7.12: Change detection with coherent scatterers for an image pair acquired over the Munich area of the "Deutsches Museum". a) SAR image from 2016, b) SAR image from 2018. c) RGB composite image with the 2018 image in red, the 2016 image in green, and the coherence in blue. d) Detected CSs for the 2016 image, e) detected CSs for the 2018 image. f) CSs color coded according to the type of change, blue: unchanged CS, green: CS only present in 2016, red: CS only present in 2018, yellow: a different CS in each date.

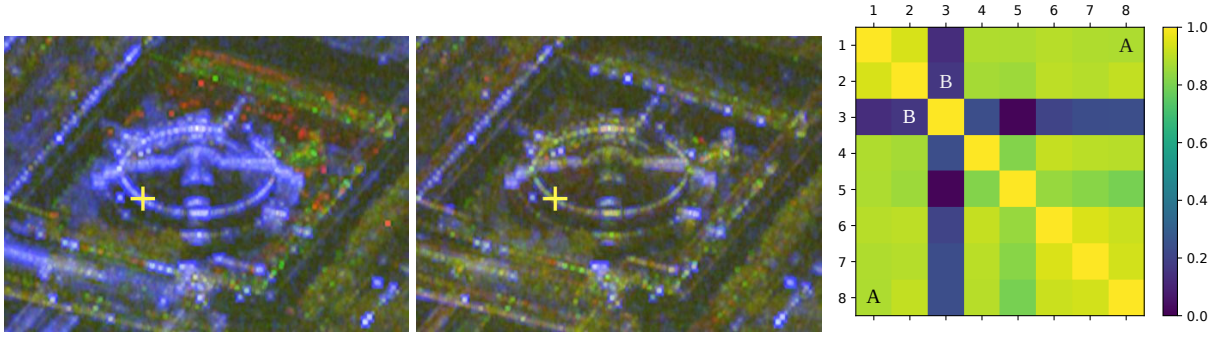
full resolution SAR images. The region shown in these images contains several buildings and two bridges, as well as some vegetation and a river. As expected, the CSs are detected in the image regions where man-made objects are located, with very few CSs being detected in the areas with water and vegetation. A multitemporal color composite image highlighting the changes between this image pair is shown in Fig. 7.12c, with both amplitude images in the green and red channels, and the coherence in the blue channel. In such a composite image, unchanged areas appear in blue and white, as they exhibit low amplitude change and high coherence. Changed areas exhibiting strong amplitude variations appear in bright green (if the amplitude decreased) and red (if it increased). Changes due to a loss of coherence with no significant amplitude variation (e.g., due to temporal decorrelation) appear in brownish and yellow colors. After applying the proposed CD method, the CSs detected in the image pair are classified into the different types of change listed in Table 5.1. The results are shown in Fig. 7.12f with the CSs color coded according to the change type. For consistency, the colors were chosen to be similar to those in the color composite image shown in Fig. 7.12c. Unchanged CSs are shown in blue, those that were only present on the first or second images are shown in green and red, respectively, and pixels containing a different CS in each image (or one that changed) are shown in yellow.

This example shows that the proposed CD approach can successfully detect the changes associated with man-made objects (e.g., in the "Deutsches Museum", the large building towards the right part of the image). Changes associated with natural targets, like the change in water level causing a strong amplitude variation at the river bank (signaled by a white arrow in Fig. 7.12c), are mostly ignored. As expected, CSs are not affected by temporal decorrelation even with a temporal baseline of approximately two years. All the CSs in the unchanged buildings and other objects exhibit high coherence and are correctly detected as unchanged. Finally, a few isolated CSs which appear to be false detections (e.g., those in the river) can also be seen. While these are very few, they could be avoided by decreasing the threshold for the CS detection. However, this would result in an increased number of false negatives (i.e., undetected CSs). Instead, these few false detections are handled during the clustering and segmentation steps.

Coherence matrix of transient changes

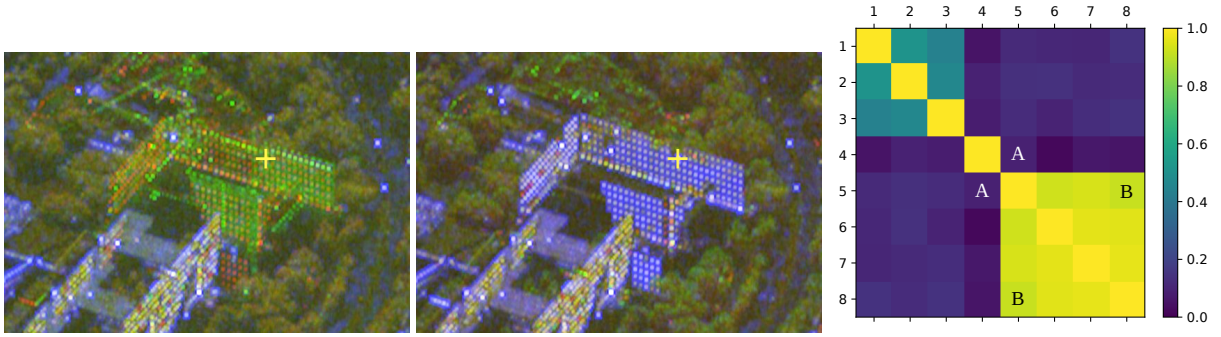
Section 5.2 introduced the concept of transient changes and briefly explained how these can be identified by their characteristic coherence matrix. To illustrate this, a real example of a transient change due to a building's roof temporarily covered with snow can be seen in Fig. 7.13. A multitemporal color composite image comparing the first and last images of a time series with eight images is shown in Fig. 7.13a. These two images were acquired almost two years apart. Some changes (highlighted in bright red and green) are visible towards the top, but the circular building in the center remains unchanged, as shown by its blue and white colors indicating a high coherence. However, in the composite image comparing the second and third images (acquired approximately one month apart) shown in Fig. 7.13b, it appears that this same circular building has changed. In this case, the coherence of the corresponding pixels is low. Visual interpretation of the full extent of the imaged scene for this third image, acquired during wintertime, suggests that snow is the reason for this low coherence. This same effect can be seen across many other buildings over the whole city, and changes in backscatter consistent with snow cover can also be seen at many other locations (e.g., at the sides of the streets). Figure 7.13c shows the coherence matrix for one of the CSs of the building (the one marked with a yellow cross on Fig. 7.13a and 7.13b). The matrix elements contain the coherence values for different image pairs: the values for the pairs shown in Fig. 7.13a (images 1 and 8) and Fig. 7.13b (images 2 and 3) are marked with an "A" and "B", respectively. For this particular example, the third image is very clearly an outlier, as all the image pairs containing this image (i.e., third column and third row) have very low coherence, whereas all the other image pairs have high coherence values.

For comparison, the change caused by the construction of a building occurring at a different location in the same time series can be seen in Fig. 7.14. Visual analysis of the time series shows that the construction work finished between the acquisition of the fourth and fifth images. These two images are compared in the color composite image of Fig. 7.14a. Strong amplitude changes highlighted in bright green and red colors can be seen across the complete façade of the building at the center of this image. Towards the top, a construction crane can also be seen in green, indicating that it was only present in the fourth image. After the fifth image, the newly constructed building remains unchanged. This can be seen in Fig. 7.14b: a composite image comparing the fifth and last images. This temporal behavior can be clearly seen in the coherence matrix, shown in Fig. 7.14c for the pixel highlighted with a yellow cross in Fig. 7.14a and 7.14b. As the building construction was finished between t_4 and t_5, the coherence $\gamma_{j,k}$ for all the image pairs with $j \leq 4$ and $k \geq 5$ is low. This behavior is clearly different from the one exhibited by the previously shown transient change. Additionally, the fact that this building then remains unchanged after t_5 can also be seen in this matrix, as $\gamma_{j,k}$ is high for all the image pairs with $j \geq 5$ and $k \geq 5$.



(a) (b) (c)

Figure 7.13: Example of a transient change: circular building temporarily covered with snow in the third image of a time series. Color composite images comparing the a) first and last images, b) the second and third images. c) Coherence matrix with the coherence values for all image pairs for the highlighted pixel. The matrix elements for the pairs in a) and b) are marked with "A" and "B".



(a) (b) (c)

Figure 7.14: Example of a lasting change due to construction work finished between the fourth and fifth images of a time series. Color composite images comparing the a) fourth and fifth images, b) fifth and last images. c) Coherence matrix with the coherence values for all image pairs for the highlighted pixel. The matrix elements for the pairs in a) and b) are marked with "A" and "B".

Change detection with a time series using CSs

Using a time series instead of an image pair introduces two additional parameters. The first one is r, related to the CD metric defined in equation 5.2. The effect of r is illustrated with an example in Fig. 7.15, using the same eight images as in Fig. 7.13 and 7.14. The resulting CD metric f_2 for detecting changes between images 2 and 3 of this series is shown in Fig. 7.15a and 7.15b for $r = 0$ and $r = 2$, respectively. The metric computed with $r = 0$ has low values (indicating change) even for unchanged buildings, whereas the metric computed with $r = 2$ correctly has high values there. This difference between both metrics is due to the previously described transient changes, caused by snow in this example. To further illustrate this, Fig. 7.15c shows a comparison of the detected CSs which remain unchanged along the complete time series according to both metrics (i.e., those with $\min_i f_i \geq \gamma_t$). The CSs highlighted in red are detected as unchanged using both $r = 0$ and $r = 2$, whereas those in blue only with $r = 2$. The blue CSs were therefore affected by a transient change at some point during the time series. For visual verification, a color composite image comparing the first and last images of this series and showing the unchanged buildings in blue can be seen in Fig. 7.15d. This example shows that higher values of r increase robustness against transient changes, as expected. When processing the complete time series with 49 images

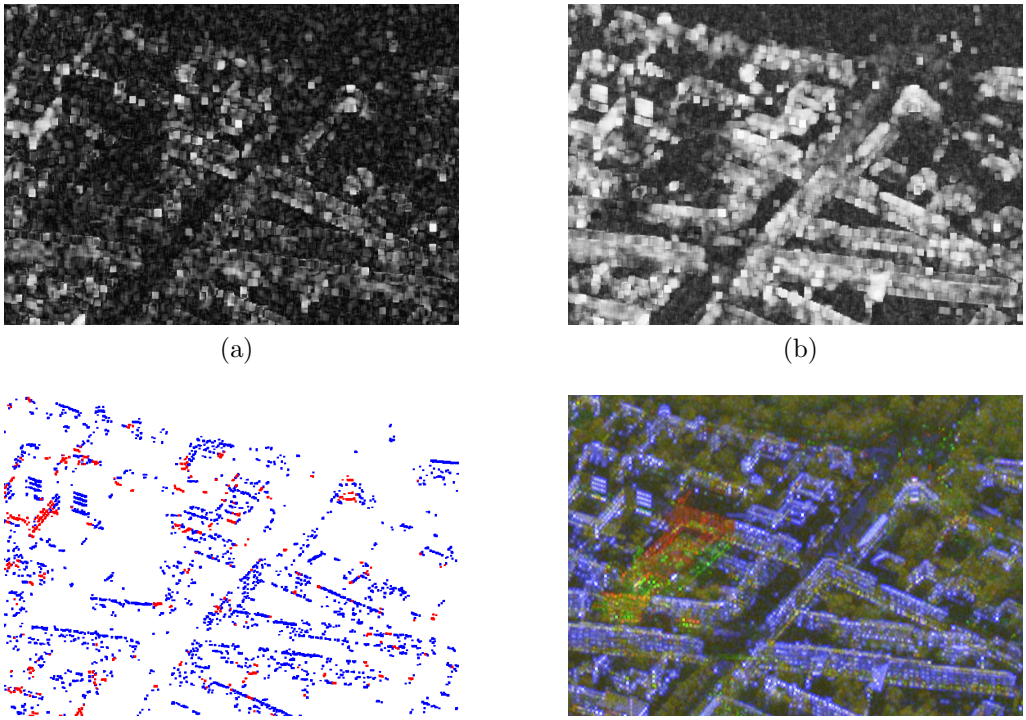


Figure 7.15: Example of the effect of the parameter r in the change detection metric defined in equation 5.2 for a time series of eight images. a) Metric f_2 computed with $r = 0$, and b) with $r = 2$. c) Unchanged CSs detected using these two metrics: red CSs detected with both, blue only with $r = 2$. d) As reference, a color composite image comparing the first and last images.

in this dataset, a value of 5 (i.e., $r = 5$) will be used for even more robustness against transient changes, at the cost of a slightly increased computation time.

The second parameter is k, related to the post-processing step for discarding CSs that are likely false positives. Experiments performed with this dataset and other TerraSAR-X data showed that a value of $k = 0.1$ appears to work well for time series of different lengths. Higher values of k result in more CSs being discarded.

Spatio-temporal clustering of CSs

The values for the parameters of the clustering step depend on the amount and density of CSs in the objects to be detected. These two factors depend mainly on the resolution of the input SLC images and the object size. Higher resolution typically results in an increased number of detected CSs. As mentioned in Section 3.2.1, the image with the detected CSs has virtually the same resolution as the input image. The lower resolution of the coherence images used for the CD should not play a significant role for the clustering, as it does not affect the number of detected CSs.

For all the examples shown here, the following parameters were used for the DBSCAN algorithm: $\varepsilon = 15$ m, $p = 20$. In this work, to filter out changes too small to be of interest, clusters containing less than 30 CSs or with a convex hull area smaller than 20 m^2 are discarded. As described in Section 5.3.1, distances and areas are computed in meters for the clustering, to compensate the different pixel spacings along range and azimuth. For all the images in this dataset, a pixel represents a ground area of around 75×17 cm.

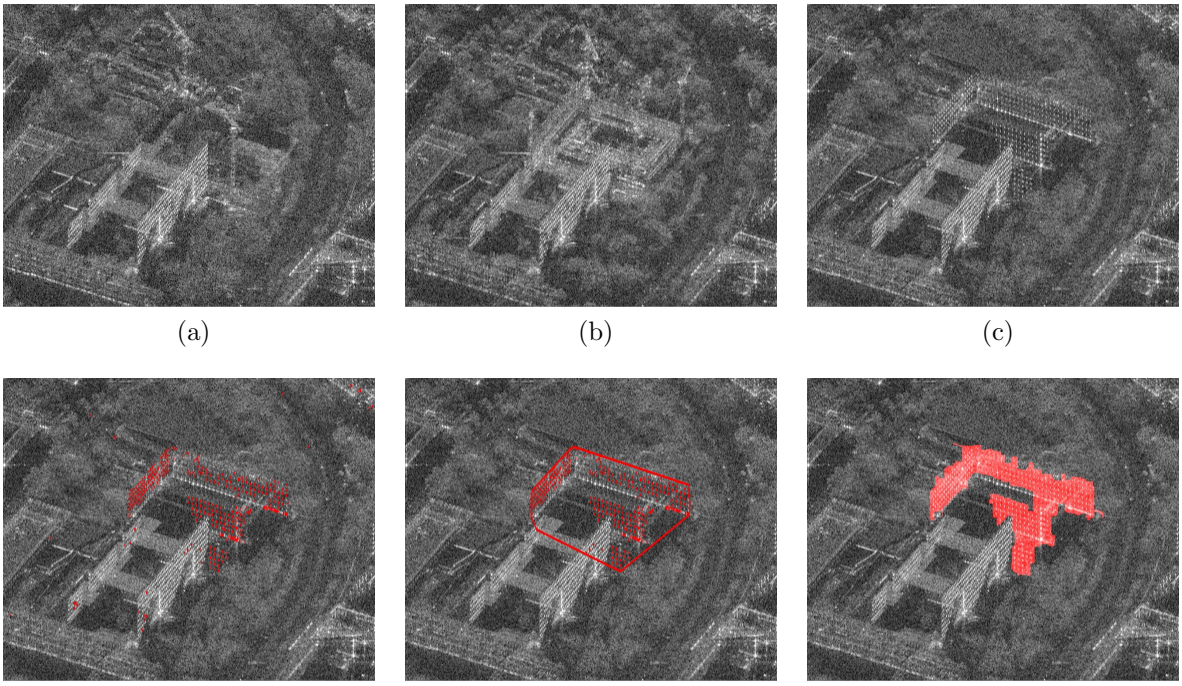


Figure 7.16: Example of object-based change detection. a-c) Three of the SAR images in a time series with eight images showing the construction of a building, d) CSs in the last image which first appeared in the sixth image, e) resulting cluster delimited by its convex hull, f) segmented change.

An example of the application of this clustering step can be seen in Fig. 7.16. Three small SAR image patches (Fig. 7.16a through 7.16c) show the construction of a building from start to finish. These are part of a time series with eight images, which were processed as described in Section 5.2. As the goal of this example is to illustrate the clustering step in a simple way, only a subset of the detected changing CSs are shown: those appearing in the sixth image and still present in the last image. Figure 7.16d shows these CSs highlighted in red over the last SAR image of the series. Most of these correspond to the newly constructed building, which was finished sometime in between the acquisition of the fifth and sixth images. The clustering results obtained with the parameters listed above can be seen in Fig. 7.16e. This example illustrates how the proposed spatio-temporal clustering can successfully group together all the CSs belonging to a changed object.

Segmentation of the detected changes

The proposed method for change segmentation has a few parameters only. The first parameter is a scaling factor, used to compute the extent of the image patches to be considered for the segmentation. In this work, this factor is set to 50%, meaning that these patches are 50% larger than the rectangles enclosing the corresponding clusters. Another parameter is the radius for the closing operation performed during the post-processing of the obtained segmentation mask. This is set to 5 pixels. The selected values seem to work well for the applications considered in this work. An example of the segmentation results obtained for a change due to the construction of a new building is shown in Fig. 7.16. The cluster with the change to be segmented and the resulting segmentation are highlighted in red in Fig. 7.16e and 7.16f, respectively. Using the temporal information obtained from the CD and the clustering results as a starting point, the proposed method achieved a rather accurate segmentation of the newly constructed building.

Additional parameters are needed for the segmentation of objects present in only one image, as the SAR amplitude must also be considered for this. The first one is a threshold Δ_A, applied to the amplitude difference in dB scale. Here, this threshold is set to $\Delta_A = 3$ dB. The second one is a threshold A_{min} related to the minimum amplitude value for pixels corresponding to man-made objects, as these tend to exhibit relatively high amplitude values. This one is set to $A_{min} = -15$ dB. Before applying these thresholds, the speckle noise should be reduced. In this work, a custom speckle filter that preserves the full resolution is applied. Multilooking is applied to the despeckled image to further reduce speckle, resulting in a 1 meter resolution in slant-range and azimuth, but keeping the original pixel spacing. This speckle filter is not described in detail, as this is out of the scope of this work. However, modern despeckling algorithms such as [Deledalle et al., 2015; Dalsasso et al., 2022] would very likely result in a better speckle reduction. Experiments have shown that these two fixed thresholds work quite well for this and other similar datasets, as they are only applied locally where changes have already been detected for many CSs. Some example results will be shown later in Section 8.2.2.

7.2.3 Practical application: monitoring of construction activity

After suitable values were selected for all the method's parameters, this method was applied to the TerraSAR-X time series of the city of Munich. For urban areas such as the one in this dataset, there are often many different changes continuously occurring across the whole imaged scene, and most general CD methods simply result in a binary change map highlighting all the changes. In contrast, the method proposed in this thesis can identify certain types of changes by their characteristic temporal behavior. To illustrate this, the proposed method was used to detect changes due to the construction of new buildings and infrastructure or renovations to existing ones. These changes were identified as described in Section 5.3.3, setting the value of ΔT set to 2 months. In addition to detecting these changes, this approach also allowed to estimate the time when the construction work was finished for each building.

Ground truth is not available for this dataset, but most of these changes can be clearly identified in the available imagery. Therefore, a qualitative evaluation of the obtained results was performed by visually verifying whether the detected changes actually corresponded to newly constructed or renovated buildings or infrastructure. The obtained results were analyzed in detail for an area around the city campus of the Technical University of Munich (TUM). Sequences of optical images showing the renovation process of the "Alte Pinakothek" and of one side of the TUM building were also used to verify the results.

In addition to changes due to newly constructed and renovated buildings, the method was also applied to detect changes caused by man-made objects appearing and disappearing in shorter time periods, which allowed to detect the build-up for different festivals. Finally, the method was also used to detect the man-made objects which remained static and unchanged during this period of almost three years. Due to the lack of ground truth, visual interpretation was also applied to qualitatively evaluate these results.

All these results will be shown in Section 8.2.

7.3 Object recognition with fully convolutional Siamese network

7.3.1 Dataset

To test the proposed method for object recognition, a dataset consisting of 60 TerraSAR-X of five different airports will be used. These five airports are denoted here with the letters from A to E. The images for each airport were acquired using different orbits and incidence angles,

Table 7.2: Number of TerraSAR-X images available for the different airports and imaging geometries.

Airport	Orbit	Look direction	Incidence angle	Number of images
A	Ascending	Right	16.1°	1
			31.1°	13
			43.0°	3
			52.1°	3
	Descending	Right	16.2°	1
			31.2°	7
			43.0°	3
B	Ascending	Right	32.5°	3
	Descending	Right	28.5°	3
			41.1°	3
C	Ascending	Right	35.9°	3
	Descending	Right	47.5°	5
D	Ascending	Right	31.1°	3
	Descending	Right	45.8°	3
E	Ascending	Right	32.6°	3
			44.6°	3

because as discussed in Section 6.1, different imaging geometries can significantly alter an object's appearance and therefore will affect its recognition. Table 7.2 provides an overview of the number of images available for each airport and imaging geometry. All these images were acquired using the Staring Spotlight imaging mode, and have a resolution of 58 cm in slant-range and 23 cm in azimuth.

In each of these images, all the instances of four different types of airplanes (denoted as #1 to #4) were labelled to create a dataset for airplane recognition. The annotations were performed manually by human operators experienced in the analysis of VHR SAR images, and verified with the help of 3-D models and SAR simulations of the different airplanes. The annotation for each airplane includes a reference to the corresponding SAR image, the bounding box defining its location in this image, the airplane type and its orientation with respect to the sensor. In total, 1614 airplanes were annotated across the 60 images. The distribution of the number of airplane labels across the five airports and the four airplane types can be seen in Fig. 7.17. The number of labels per airplane type, which can be seen in Fig. 7.17a, shows a small class imbalance, with many more samples available for the airplane type #1. In the same way, Fig. 7.17b shows that there are more samples available for some airports than others. A more detailed breakdown can be seen in the 2-D histogram of Fig. 7.17c, where it can be seen that certain airplane types are only present at specific airports (e.g., airplane #2 is only present at airport A).

Other types of airplanes present in the available images were also annotated with bounding boxes, but without any additional information (e.g., type was annotated as "other"). These samples cannot be used for classification due to the lack of information, but they can be used as negative samples during training.

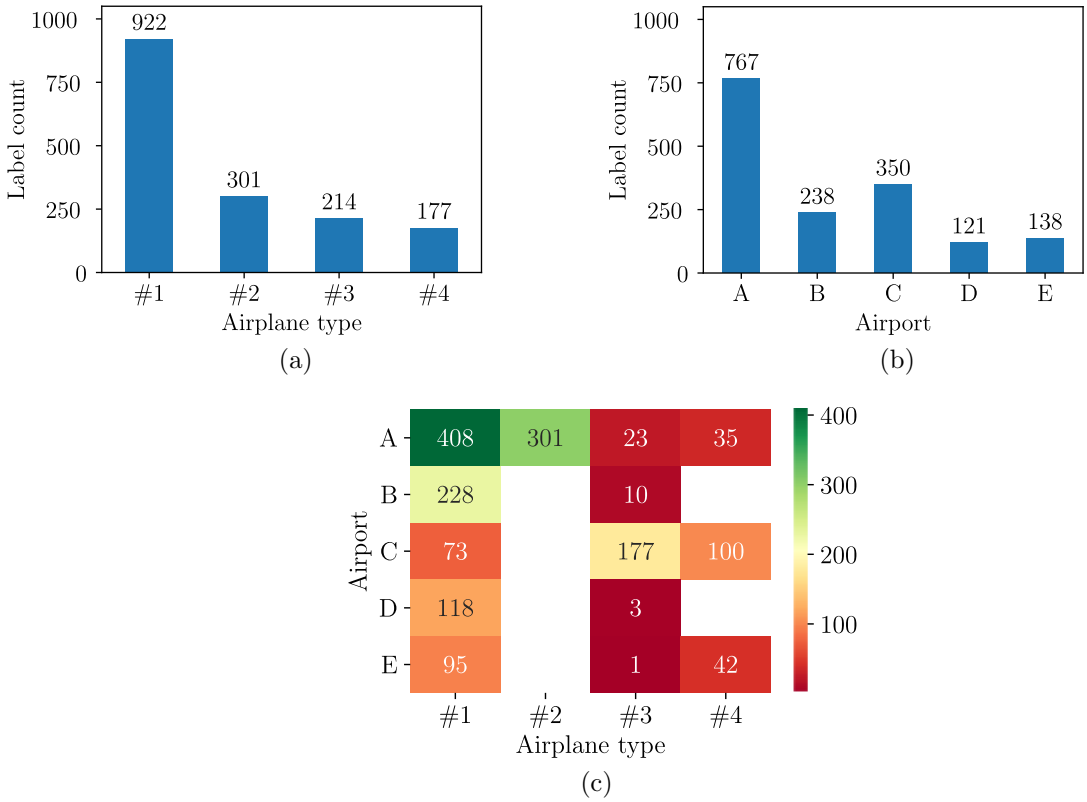


Figure 7.17: Distribution of the annotated airplanes in the dataset for the different object classes and locations. a) Number of labelled airplanes for each class. b) Number of labelled airplanes at the different airports. c) Number of labelled airplanes for each class and airport.

Because the proposed approach using a Siamese network will only consider two objects as similar if they are of the same type and have a similar imaging geometry, it makes sense to analyze the available imaging geometries for each of the four airplane types. This can be done by dividing the range of possible incidence angles (e.g., 15° to 55°) and possible object orientations (0° to 360°) into multiple discrete intervals (i.e., bins). Then, the labelled airplane samples can be assigned to the corresponding bins to generate a 2-D histogram. The resulting histograms for the four airplane types can be seen in Fig. 7.18 through 7.21. For these visualizations, a bin size of 5° was chosen for the incidence angle, and of 10° for the airplane orientation. Here, it can be seen that the samples are irregularly distributed across the different incidence angles and object orientations, with a high number of samples for certain imaging geometries and very few or none for many others. The available incidence angles are given by the used TerraSAR-X images (i.e., those listed in Table 7.2), and the amount of samples for a given incidence angle can be increased by simply acquiring more images. On the other hand, the sampling of the different orientations for a given object cannot be so easily controlled, as this depends on the actual orientation of the objects in the imaged scenes with respect to the SAR sensor's line of sight (i.e., range). A given scene can be imaged from different directions by using different orbits (e.g., ascending and descending) and look directions (left and right looking), but the number of possibilities is typically limited for most spaceborne SAR missions. Also, the orientation of the imaged objects cannot be controlled, and some orientations can simply be more common for certain objects (e.g., airplanes are often parked at specific positions and with the same orientations).

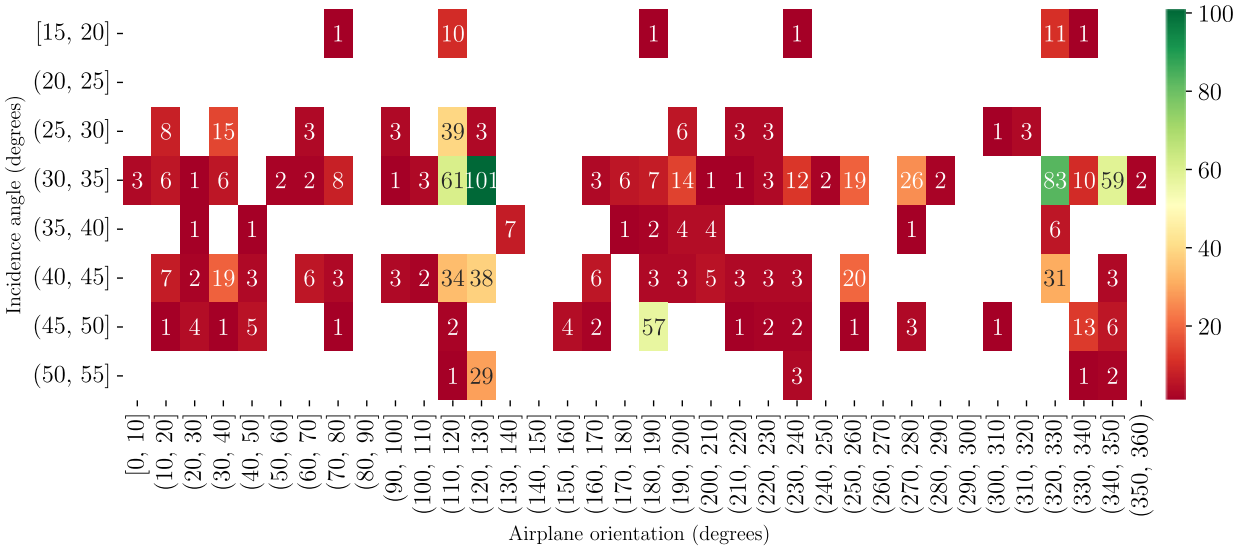


Figure 7.18: Number of labels for airplane type #1 for the different incidence angles and object orientations.

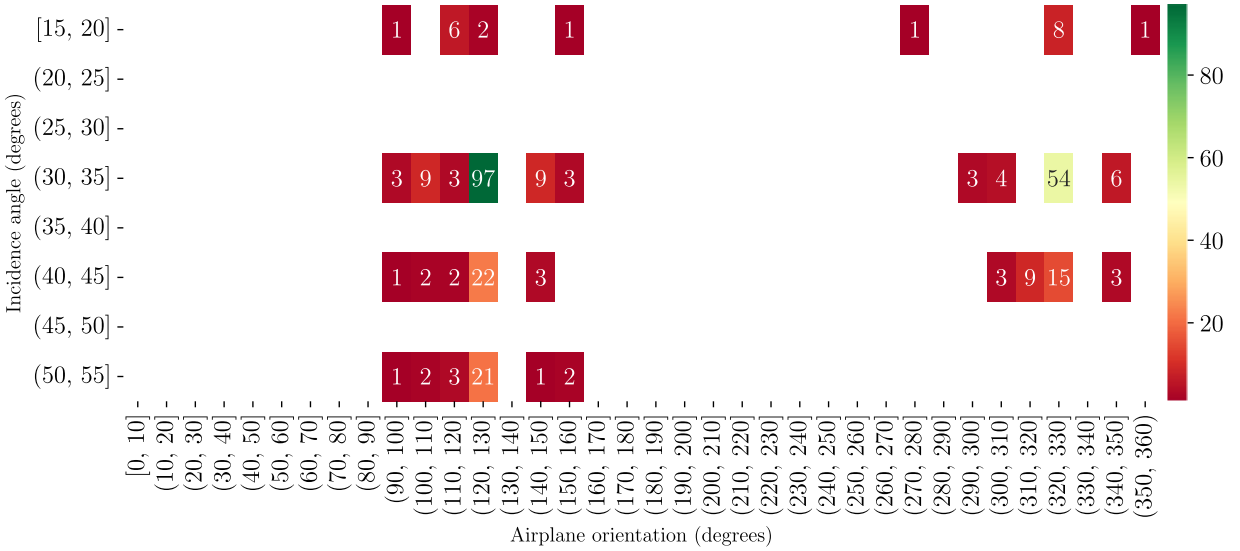


Figure 7.19: Number of labels for airplane type #2 for the different incidence angles and object orientations.

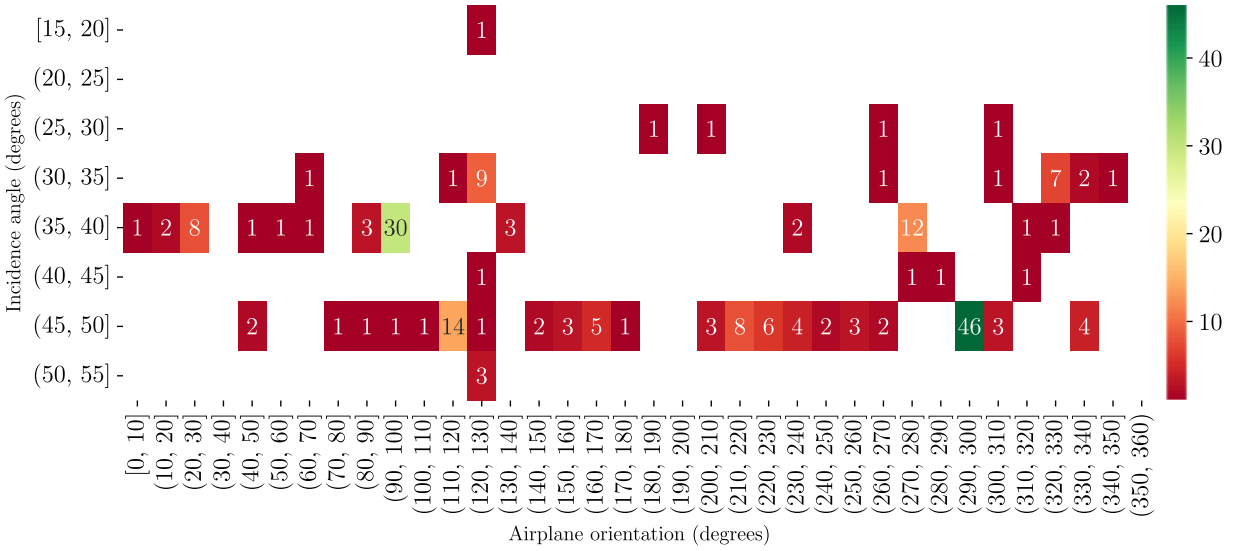


Figure 7.20: Number of labels for airplane type #3 for the different incidence angles and object orientations.

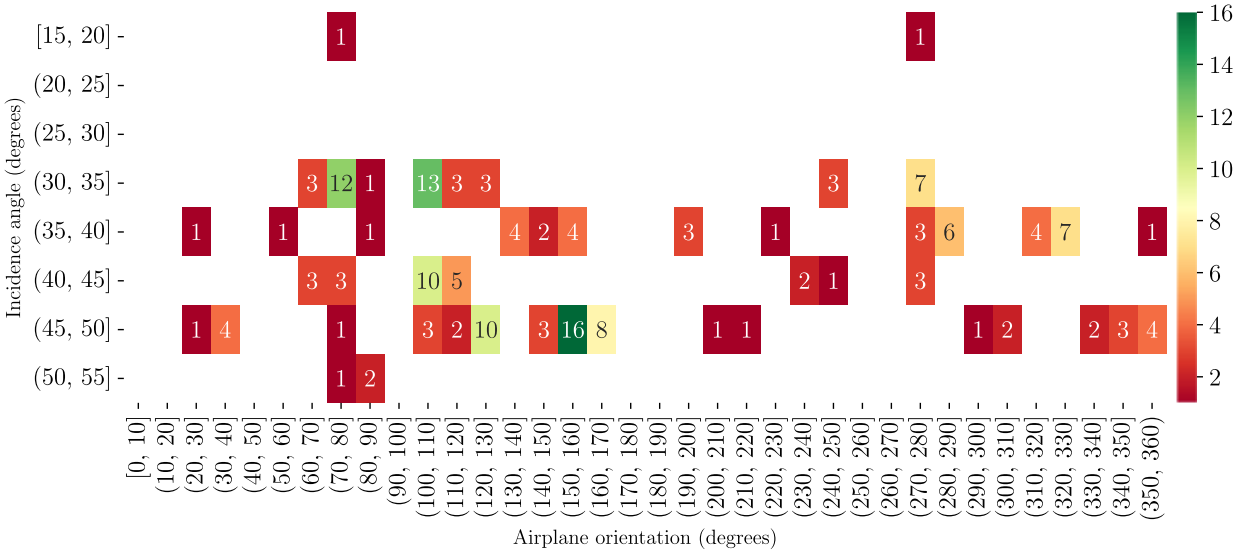


Figure 7.21: Number of labels for airplane type #4 for the different incidence angles and object orientations.

Negative generation

As explained in Section 6.3.2, negative samples (i.e., not containing any of the objects in the dataset) are useful during training. A method was proposed to automatically generate negative examples in a controlled way, as a purely random negative generation can result in a large amount of easy negative samples (e.g., homogeneous clutter areas like vegetation, water, etc.) which are not very valuable for training a CNN. With the proposed criteria, three types of negative samples can be generated. The first type will contain many CSs, and will therefore contain other man-made objects. The second type will have less CSs but a high coefficient of variation, and will most likely exhibit different types of surfaces and maybe some man-made objects as well. The third type will have very few CSs and a low coefficient of variation, and will typically correspond to relatively homogeneous clutter areas. Using this approach, 150 negative samples were generated for each of the images in the dataset, with 60% of them corresponding to the first type, 35% of them to the second type, and the remaining 5% to the third type. Some of the generated negative samples for one of the TerraSAR-X images can be seen in Fig. 7.22. Figures 7.22a through 7.22c correspond to the first type of negatives, Fig. 7.22d through 7.22f to the second type, and Fig. 7.22g through 7.22i to the third one.

Definition of different test cases

Rather than directly applying the proposed method to the complete dataset, different test cases with increasing difficulty and corresponding to different subsets of the dataset will be defined:

- One airplane type (#1) in a single airport (A), with 408 airplane samples across 31 images.
- One airplane type (#1) in five airports, with 922 airplane samples across all 60 images.
- Four airplane types in a single airport (A), with 767 airplane samples across 31 images.
- Four airplane types in five airports, full dataset with 1614 airplane samples across 60 images.

For the cases where only one airplane type should be detected, the labelled samples corresponding to the other airplane types are added to the list of negative samples.

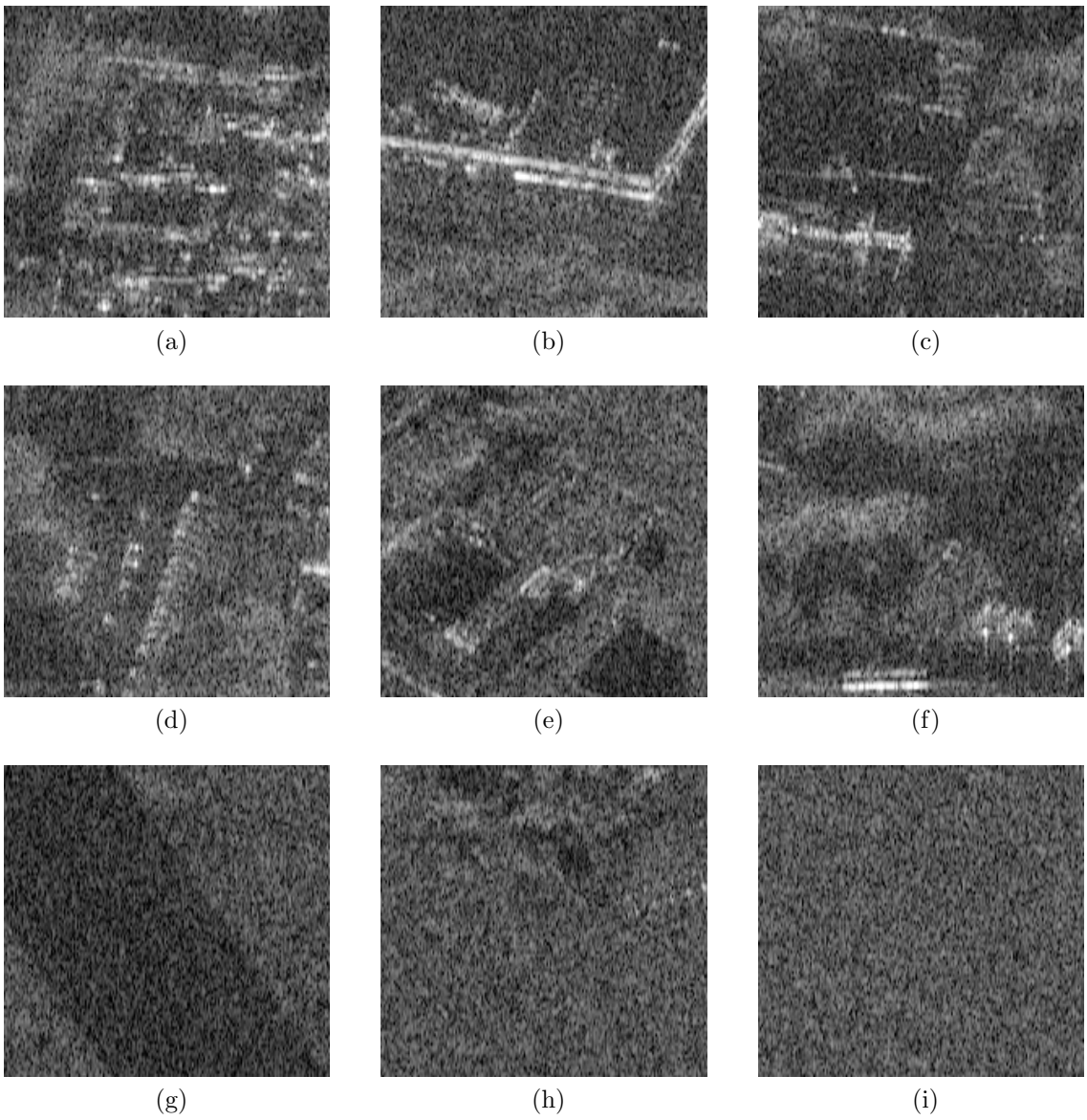


Figure 7.22: Examples of automatically generated negative training samples. Different types of negatives are generated using different criteria: a-c) samples with a high number of CSs, d-f) samples with less CSs but a high coefficient of variation, g-i) samples with very few CSs and a low coefficient of variation.

To train and evaluate the proposed network architecture in each of these test cases, the corresponding data still needs to be split into training, validation and testing. The training data is used to train the network as described in Section 6.3. During training, the network performance is periodically evaluated on the validation data, to monitor how the network is learning and determine when training should be stopped. After the training is finished, the test data is used for the final performance evaluation. In this thesis, the data will be split on an image level, with each SAR image and all the corresponding annotations assigned to either the training, validation or test set. The split could also be done on a tile level by dividing the SAR images into tiles first, which would allow a finer control on the number of samples assigned to each set. However, the split on an image level is closer to the real application scenario (where the method needs to be applied to detect objects on the newly acquired SAR images), which will make the obtained results more representative.

In an ideal application scenario, the proposed method would be applied to images which are similar to those used for training. However, it should also be possible to apply the method to images of different locations, acquired using a different orbit or incidence angle, or during different seasons. As previously described in Section 6.1, these different conditions can significantly alter the appearance of the objects in the SAR images, which will most likely affect the method's performance. To evaluate how much the performance of the proposed method drops when certain conditions are different in the training and test data, different splits will be generated for each scenario. Here, the following splits will be created:

- Varied split: the test images have similar characteristics to those used for training, representing the ideal case. Approximately 70% of the images, including all the possible conditions (e.g., imaging geometries, seasons...), are used for training. 10% of the remaining images are used for validation, and 20% for testing. This type of split is applied to all the different test cases.
- Incidence split: the test images have incidence angles that are not available in the training set. To analyze the effect of different incidence angles in isolation, without the influence of seasonal changes or different locations, this split is applied only to the test cases with a single airport, and the images with snow (the most significant seasonal change) are excluded. All the images with an incidence angle of 43° are used for test, and the rest (with 16°, 31° and 52° incidence) for training and validation.
- Season split: all the winter images (most of them with snow) are used for test, and the rest for training and validation. To analyze the effect of seasonal changes in isolation, without the influence of different incidence angles or locations, this split is applied only to the test cases with a single airport, and only the images with a 31° incidence angle are used.
- Airport split: all the images of airport E used for test, and those of airports A to D are used for training and validation. For obvious reasons, this split is only applied to the test cases with multiple airports.

In all cases, the images used for validation have the same characteristics as those used for training, so that the network cannot be optimized for the different conditions of the test data in any way. Because the test scenarios with one or multiple airplane types have the exact same images, the same splits will be applied for both. The number of images used for training, validation and testing for the different splits are listed in Table 7.3. In the same way, the number of training, validation and test samples for each airplane type in each split can be seen in Table 7.4. For the test cases with a single airplane type, the number of samples can be read from the "#1" column, as the splits are identical.

Table 7.3: TerraSAR-X images used for training, validation and testing in the different test cases.

Test case	Split	Train	Val	Test	Total
Single airport	Varied	21	3	7	31
	Incidence	19	3	6	28
	Season	14	2	4	20
Multiple airports	Varied	40	7	13	60
	Airport	49	5	6	60

Table 7.4: Number of training, validation and testing samples for each class in the different test cases.

Test case	Split	Number of samples for each class (train/val/test)			
		#1	#2	#3	#4
Single airport	Varied	275/40/93	204/30/67	13/1/9	25/3/7
	Incidence	255/40/74	186/28/60	19/0/2	23/3/6
	Season	196/26/51	135/19/37	14/1/2	18/2/4
Multiple airports	Varied	617/114/191	204/30/67	128/31/55	115/24/38
	Airport	743/84/95	282/19/0	185/28/1	123/12/42

7.3.2 Hyperparameter selection and experiments training the network

To test the proposed method with this dataset, the network architecture described in Section 6.2 was implemented using the PyTorch Lightning framework. Three popular CNN architectures were considered for the feature extraction backbone: AlexNet [Krizhevsky et al., 2012], ResNet [He et al., 2016] and ConvNeXt [Liu et al., 2022]. Due to the limited amount of data available, some of the smaller variants of ResNet (ResNet-34) and ConvNeXt (ConvNeXt Tiny) are used. The impact of these different CNN architectures for feature extraction in the overall performance will be analyzed and illustrated below.

To train this network, the values of many hyperparameters (i.e., parameters that cannot be learnt from the data during training) must be first set. The effect that all these hyperparameters have on the method's performance can be empirically analyzed, but this requires training the network several times for the different hyperparameter settings. A comprehensive study on the effect of all hyperparameters will be avoided here, as training a deep CNN is a relatively time consuming process, and doing that many training runs would require a large number of GPU hours and consume a significant amount of energy. Here, the initial values for all the hyperparameters were chosen based on an educated guess. Then, a few experiments were performed to improve this initial selection, by trying to optimize the performance in the validation set of some of the aforementioned test cases. Below, the selected values for all the hyperparameters will be listed, and a few of the performed experiments will be shown.

The network is trained using stochastic gradient descent (SGD) with momentum in a single GPU (an Nvidia RTX 3090). Different learning rate values l_r will be used for the different experiments. However, the same learning rate schedule is always applied: first, a warm-up learning rate of $l_r/4$ is used during the first epoch, then the chosen learning rate value l_r is used until the tenth epoch, and afterwards an exponential decay with an exponent of 0.95 is applied. Momentum is set to 0.9 and weight decay is not applied. Half-precision floats (i.e., 16-bit floats) are used to speed up training. This also allows using larger batch sizes, which requires fitting more data into the GPU memory.

The proposed similarity criterion has two parameters, δ_θ and δ_α, which are used in equations 6.1 and 6.2 to determine whether two samples have a similar imaging geometry. Here, these are set to $\delta_\theta = 20°$ and $\delta_\alpha = 30°$, as these values have empirically shown to perform well.

During the pre-processing of the SAR images, their dynamic range was limited by applying equation 6.3 with the following values: $A_{min} = -30$ dB and $A_{max} = 20$ dB. The effect of applying a speckle filter during pre-processing was also analyzed, and the results of this experiment will be shown and analyzed below.

Regarding the sampling of the training data, the templates are sampled using the partially balanced sampling approach described in Section 6.3.2. The probabilities of sampling the different airplane types are computed by applying equation 6.4 with $a_{cls} = 0.5$, and the probabilities of sampling the different imaging geometries within a given type are computed with $a_{geo} = 0.25$. When sampling the search images to form the input image pairs, these will be sampled with the probabilities $p_p = p_n = 0.5$ (i.e., 50% of positive and 50% of negative pairs). 60% of the negative pairs will be generated using negative background samples, 30% using samples of airplanes of a different type, and the remaining 10% with using samples of the same airplanes but with different imaging geometry. The size of the sampled templates is rounded up to a multiple of 64 pixels, with a minimum size of 128 pixels, and the size of the search images is 3 times the size of the corresponding template. These image pairs are organized into batches, using a batch size of 32. For each training epoch, 2048 batches are randomly sampled with the probabilities listed above.

As described in Section 6.3.2, no complex data augmentation strategy was implemented. Here, the bounding box of the sampled search images is shifted by applying a random translation (of up to ±12.5% of its size) before reading the corresponding image patches. This random translation does not only serve as a data augmentation strategy, but it also allows to use deeper CNNs for feature extraction as previously discussed. The suggested azimuth flip is not applied here, as it did not seem to provide a significant benefit during the initial tests.

Finally, the ground truth for the two output branches is generated as described in Section 6.3.3, using a value of $k = 0.125$ in equations 6.5 and 6.6. The focal loss applied to the output of the classification branch is computed using a weighting factor $\alpha = 0.75$ (to give more weight to the less common positive samples) and an exponent $\gamma = 2$. The losses of the two branches are combined using equation 6.8 with a weight of $\lambda = 0.1$.

Effect of separate BatchNorm layers in the two branches of the Siamese CNN

One of the modifications proposed in Section 6.2 over the original SiamFC++ architecture [Xu et al., 2020] was to use different BatchNorm layers for the two branches of the Siamese CNN, to address the fact that the template and search images will have significantly different statistics. Here, the impact of this modification will be briefly analyzed.

For this experiment, the proposed network was trained twice: once using a purely Siamese CNN (with the same BatchNorm layers for both branches) and once with the suggested modification. The feature extraction is performed using the same CNN architecture as in the original SiamFC paper by Bertinetto et al. [2016], which is similar to the convolutional stage of AlexNet, but without padding and with a lower stride. The previously described learning rate schedule is used, with $l_r = 2 \times 10^{-4}$. Other than that, the previously listed values were used for all the remaining hyperparameters. The data from the "varied split" of the first test case (with only one airplane type in one airport) was used in this experiment. Here, no speckle filter was applied to the SAR images during pre-processing.

To avoid any differences in both training runs caused by the random sampling of the training and validation data, the same seeds were used to initialize the random number generators. This

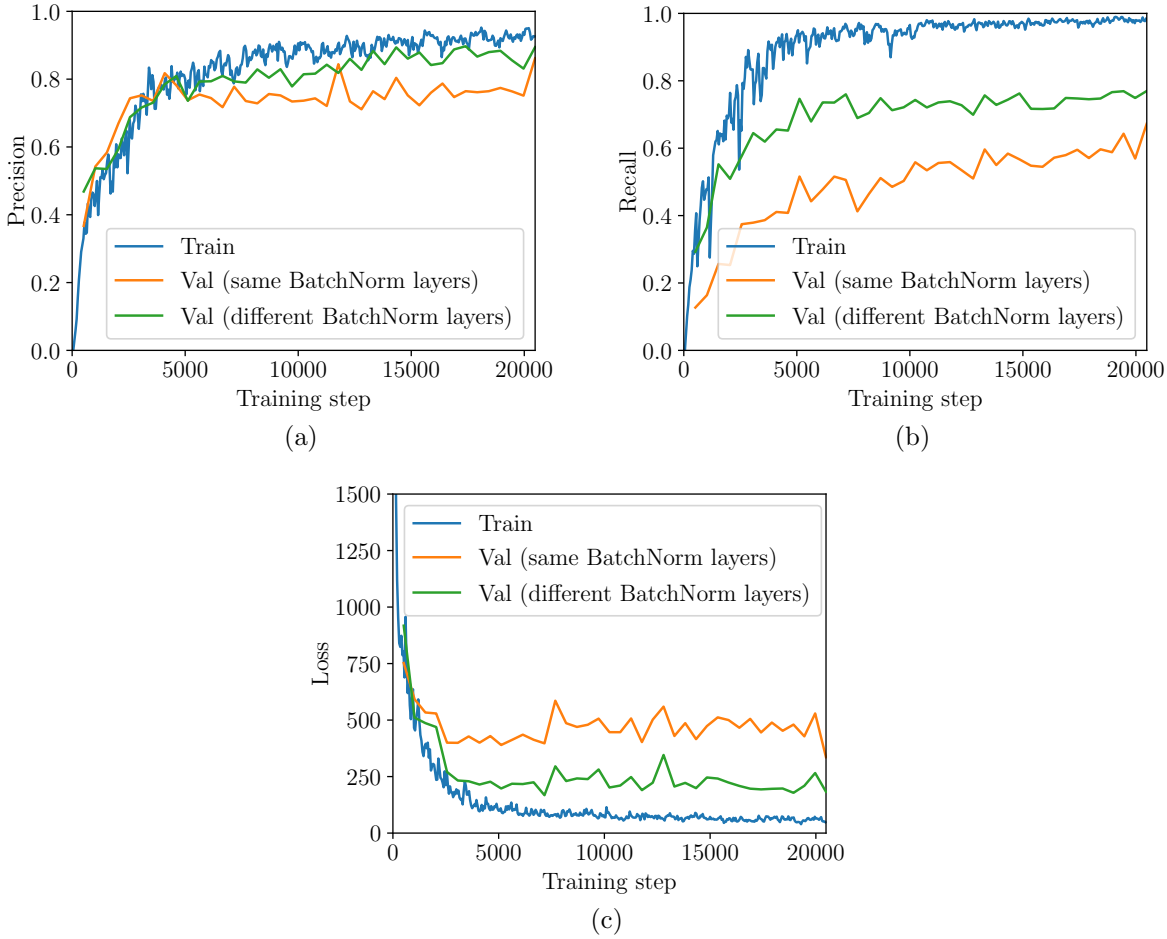


Figure 7.23: Effect of using different BatchNorm layers in the two branches of the Siamese CNN, shown for multiple metrics for the validation set at different points during training. The corresponding metrics for the training set are not affected by this change, and are only shown as a reference. Metrics computed for all the pixels in the output similarity map: a) precision, b) recall. c) Total loss value.

results in a deterministic behavior, which can be verified by the fact that the training loss is identical for both runs. An identical training loss is expected even if the network design was slightly modified, as the effect of using separate BatchNorm layers can only be observed during validation or testing and not during training.

A comparison of the results obtained with both approaches can be seen in Fig. 7.23. For this analysis, different metrics are computed for 4096 randomly sampled pairs of the validation set. The precision and recall computed for all the pixels in the output similarity map are shown in Fig. 7.23a and 7.23b, and the total loss is shown in Fig. 7.23c. It is important to note that the pixelwise precision and recall shown here do not correspond to the actual detection performance, which requires applying the post-processing described in Section 6.4, and which will be analyzed later. In this case, the recall measures the percentage of the positive pixels (in the ground truth similarity map) that are correctly identified (i.e., with a predicted similarity score above 0.5). The precision measures the percentage of the predicted positive pixels that are actually positive (in the ground truth). In summary, a high recall indicates a low number of false negatives in the pixelwise similarity map, and a high precision a low number of false positives. To show how these metrics evolve as the training progresses, they are evaluated every 512 training steps (i.e., 4 times per epoch). As a reference, the same metrics are shown for the training data, averaged over every 50 training steps. Exponential smoothing with a smoothing factor of 0.5 is applied to the training

metrics, as these are quite noisy. The plots in Fig. 7.23 show that the proposed modification is clearly beneficial, as it leads to a significantly lower validation loss, as well as a higher pixelwise precision and recall, without increasing the network's complexity. Similar improvements can be seen also on the test set or when using the data from a different test case. Because of this, different BatchNorm layers will be used in the two branches of the Siamese CNN for all the successive experiments.

Effect of speckle filtering

As suggested in Section 6.3.1, a speckle filter should ideally be applied to all SLC SAR images during pre-processing to reduce the effect of speckle noise, which is likely to affect the detection performance. Here, the impact of speckle reduction will be briefly analyzed. This experiment is performed in the exact same way and with the same settings as the previous one, but here different pre-processing methods were applied to the dataset before each training run. The results obtained with the original SAR amplitude images will be compared to those obtained when applying a custom speckle filter, as well as with those obtained when using a binary image with the detected CSs. An example of the three different types of input images used for this comparison can be seen in Fig. 7.24. The used speckle filter, shown in Fig. 7.24b, greatly reduces the speckle noise at the expense of a slightly worse spatial resolution (which was reduced to 1 meter in both slant-range and azimuth). The CS detection, whose results are shown in Fig. 7.24c, completely eliminates the speckle as well as all the background information, leaving only the most prominent scattering centers of the objects in the scene.

The obtained results for the three different inputs can be seen in Fig. 7.25, which shows the same metrics used in the previous experiment. The results for the despeckled image (shown in orange) are consistently better across all three metrics. In this experiment, the used speckle filter introduces an improvement of about 5% in both the pixelwise precision and recall. On the other hand, using the binary images with the detected CSs results in a similar or even slightly worse performance than the original amplitude image. This could be due to the fact that the CS detection removes some valuable information (e.g., the radar shadow) in addition to the speckle noise and the background, or maybe the chosen CNN architecture is simply not well suited for this kind of sparse data.

Based on these results, the speckle filter shown in Fig. 7.24b will be used during pre-processing for all the following experiments. An extensive comparison of different speckle filters is outside of the scope of this work, but it is to be expected that state of the art despeckling methods such as those proposed by Dalsasso et al. [2022] or Deledalle et al. [2015] bring some additional performance gains.

(a) (b) (c)

Figure 7.24: Comparison of different pre-processing methods applied to the SAR image of an airplane. a) Original SAR amplitude image. b) Despeckled image. c) Coherent scatterers.

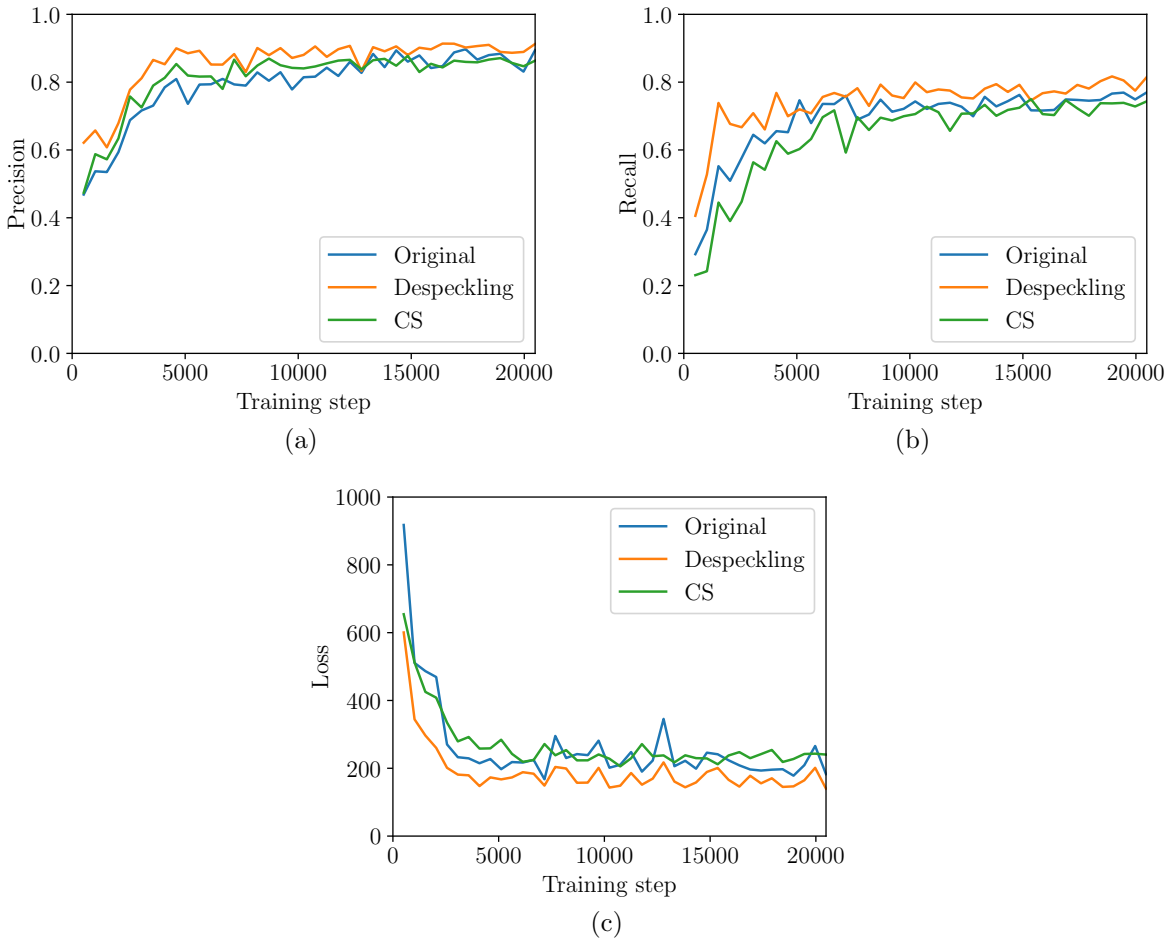


Figure 7.25: Effect of applying different pre-processing methods to the input SAR images, shown for multiple metrics for the validation set at different points during training. Metrics computed for all the pixels in the output similarity map: a) precision, b) recall. c) Total loss value.

Choosing a CNN for feature extraction

Here, the effect of using different CNN architectures for feature extraction will be analyzed. For this experiment, the results obtained using the previously described variation of AlexNet (denoted here as AlexNet*) will be compared to those obtained using the popular ResNet-34 and the more recent ConvNext Tiny (denoted here as ConvNeXt-T). In all cases, only the convolutional stage of these CNNs is used, and the fully connected layers used for classification are removed. These three resulting networks have 2.4 (AlexNet*), 21.3 (ResNet-34) and 27.8 (ConvNeXt-T) million parameters. Due to the size of ResNet-34 and ConvNeXt-T and the limited amount of data available in the used dataset, pre-trained versions of these networks will be used here. Specifically, the versions available in the PyTorch library (pre-trained on ImageNet [Deng et al., 2009]) are used. Because these pre-trained networks expect RGB input images, the single channel SAR images are duplicated and stacked to generate RGB images with three identical channels. As the used AlexNet* network is slightly different from the original AlexNet network, no pre-trained version was available. However, due to the reduced number of parameters, this network is easier to train from scratch.

To properly evaluate the impact of deeper networks, this experiment will be performed using more challenging data: the "varied split" of the fourth test case (the four airplane types in all five airports). Due to the larger amount of data and different CNN architectures, the previously

used learning rate was modified for this experiment. Here, a learning rate of $l_r = 4 \times 10^{-3}$ is used, with the same schedule as before. The pixelwise metrics which were analyzed in the previous experiments cannot be compared here across the different CNNs, as they have different strides, resulting in outputs of different sizes. Besides, these pixelwise metrics cannot show the different performances for each airplane type. Instead, the complete processing chain (including post-processing) is applied as described in Section 6.4 to all the images in the validation set. For this, a template database is generated as previously described, randomly selecting one training sample for each bin, using a bin size of 5° for the incidence angle and of 10° for the object orientation. This results in a set of bounding boxes with the detections for each of these SAR images, which are compared to the ground truth. The results are evaluated using the popular average precision (AP) metric for object detection [Padilla et al., 2020]. This metric measures the area under the precision-recall curve (which describes the object detection performance for all the possible thresholds), and an AP of 1 indicates a perfect detection performance. The AP is computed separately for each airplane type, and a mean AP value is also computed by averaging the results for the different types. An IoU threshold of 0.33 is used when applying NMS, and a threshold of 0.5 is used to evaluate whether the resulting bounding boxes are correct. A detailed description of all these object detection metrics can be found in [Padilla et al., 2020]. This is repeated after each training epoch, to illustrate how the performance improves for the different networks as the training progresses.

The results of this experiment can be seen in Fig. 7.26. As expected, the deeper networks perform better, especially for the airplanes of types #3 and #4, which appear to be more difficult to recognize. This results in a significant improvement to the mean AP values. The performance of the two deeper networks is quite similar, but the best mean AP value (0.952) is obtained with ConvNeXt-T. For comparison, the best mean AP value obtained with ResNet-34 is 0.919. In addition to this, the plots for airplanes #3 and #4 for ConvNeXt-T finish in an upward trend, indicating that the performance might continue to improve slightly during a few more epochs. Because of this, the ConvNeXt-T network will be used for the feature extraction in all the following experiments performed in this thesis.

This experiment has clearly shown the benefit of using a deep CNN for feature extraction. Here, pre-trained versions of these deep networks were chosen, due to the relatively small size of the available dataset. For completeness, the impact that this pre-training has in the performance will also be analyzed here. For this, the network was trained once again, but initializing the ConvNeXt-T CNN with random weights (i.e., without pre-training). The obtained results, which can be seen in Fig. 7.27, demonstrate the importance of ImageNet pre-training, as without it the best mean AP value drops from 0.952 to 0.652.

Sampling of the training data

As a final experiment, the impact of the proposed strategy for sampling the training data will be analyzed. For this, the ConvNeXt-T network from the previous experiment will be trained multiple times, each time using a different strategy for sampling the training data, and the obtained results will be compared. The proposed sampling strategy, described in detail in Section 6.3.2, involves using both sampling the templates in a partially balanced way (controlled by the hyperparameters a_{cls} and a_{geo}), and generating a certain amount of positive and negative pairs (controlled by the hyperparameter p_p). Here, the network will be trained for different combinations of these three hyperparameters. As a reference, the network will be first trained using the proposed values: $a_{cls} = 0.5$, $a_{geo} = 0.25$ and $p_p = 0.5$. To see the benefit of sampling the templates in a partially balanced way, the network can be trained while sampling the templates with their original probabilities (i.e., by setting $a_{cls} = a_{geo} = 0$). Finally, to see the impact of

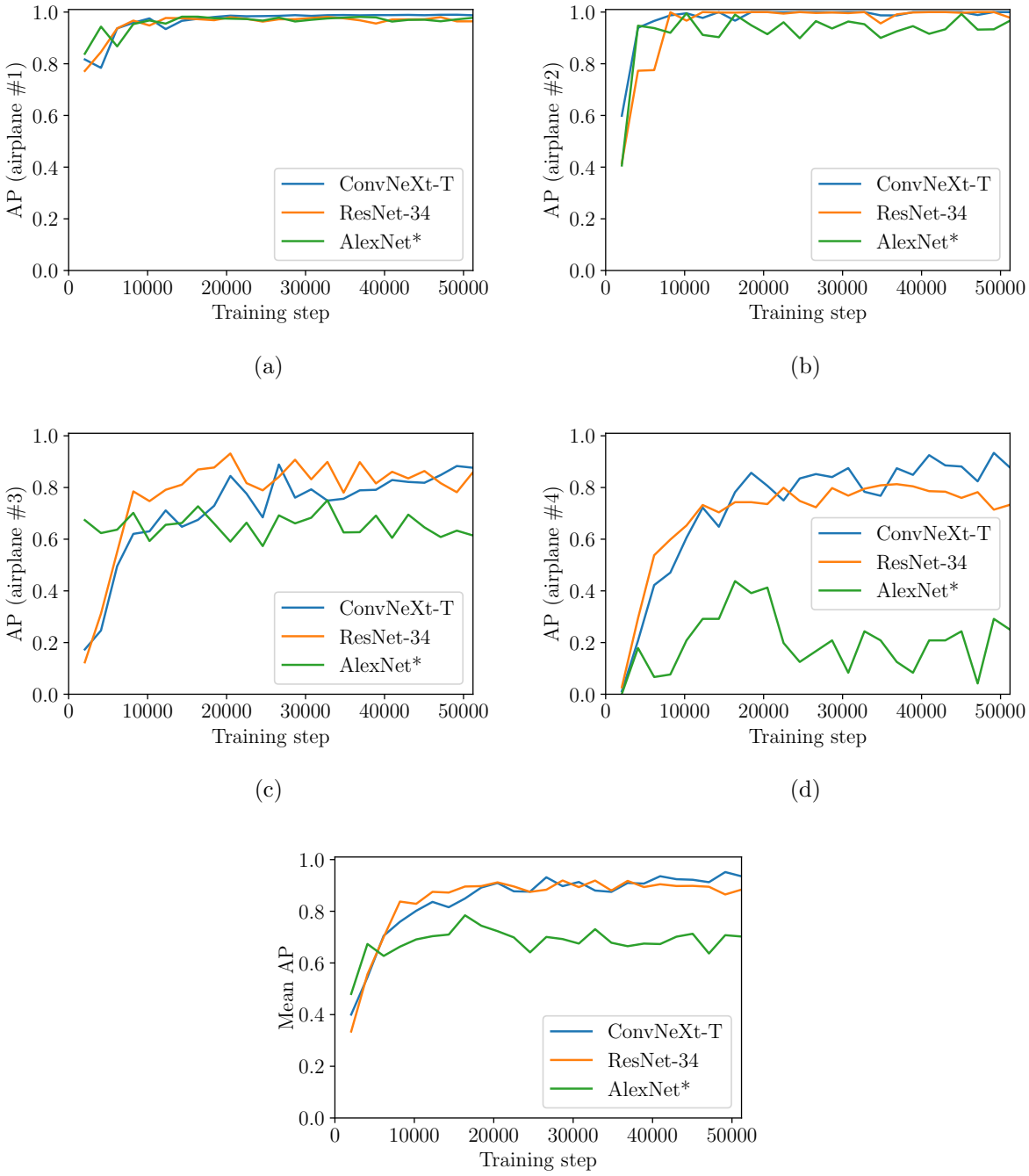


Figure 7.26: Effect of using different CNN architectures for the feature extraction in the average precision for the images of the validation set. Results for the different airplane types: a) #1, b) #2, c) #3, d) #4. e) Mean average precision.

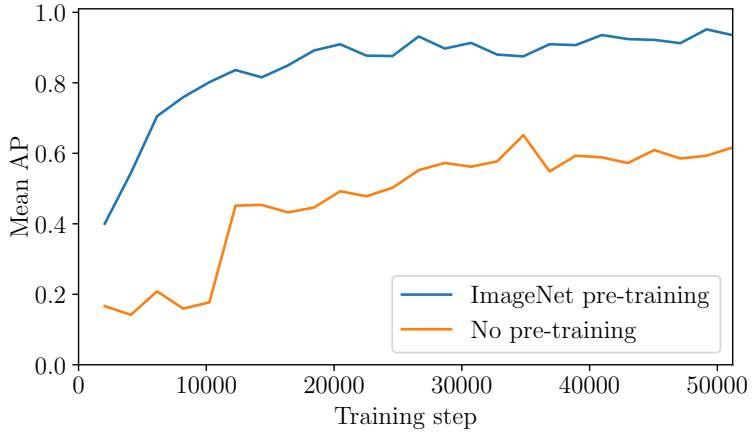


Figure 7.27: Effect of pre-training on ImageNet when using a deep CNN for feature extraction, measured in the mean average precision for the images of the validation set.

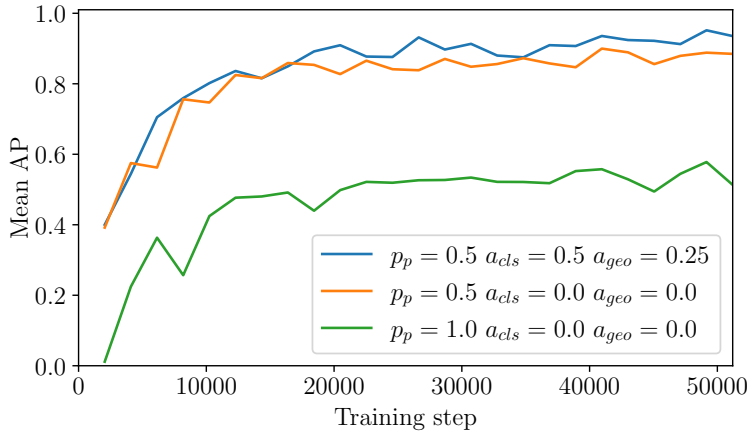


Figure 7.28: Effect of different strategies for sampling the training data in the mean average precision for the images of the validation set. The different strategies correspond to different settings for the hyperparameters p_p, a_{cls} and a_{geo}.

using both positive and negative pairs, the network can be trained once again using only positive pairs (i.e., with $p_p = 1$), as is typically done in the original task of video tracking for which this kind of network was designed.

The obtained results for these three sampling strategies can be seen in Fig. 7.28, which shows the mean AP across all the images in the validation set computed after every training epoch. These results clearly show that the proposed sampling strategy is beneficial. The most important aspect is the generation both positive and negative pairs for training the network: the training run with $p_p = 1$ (i.e., the green curve) results in a much lower AP than those with $p_p = 0.5$. Using only positive pairs results in a much worse performance because the network is only trained to distinguish an object from its immediate surroundings. The partially balanced sampling of the templates results in a smaller but still significant improvement to the mean AP: the best mean AP increases from 0.9 to 0.95. This is mostly caused by a larger improvement to the AP for airplane type #4, which improves from 0.7 to 0.93. This seems to indicate that a partially balanced sampling of the templates can be very beneficial for the less common and/or more difficult object classes.

7.3.3 Practical application: monitoring of airport activity

The previous experiments suggest that the proposed ATR method, with the chosen network design and all the previously listed hyperparameters, can perform very well for the task of airplane recognition. Such a method could be applied to SAR time series to determine how many airplanes, and of which types, are present in a given airport at different times. This information can be relevant for security-related applications and can also provide an insight into economic activity. To evaluate the suitability of the proposed method for this kind of application, its performance was analyzed using all the previously defined test cases. A comprehensive analysis of its generalization capabilities was conducted, as in a real application scenario such a method will eventually be applied to images which are significantly different from those used in training. For this analysis, the different splits for each test case were used to understand how much the performance drops under conditions that are not available in the training set.

For all the test cases, the network was trained using the hyperparameters previously listed in Section 7.3.2. Also, all the design choices that proved beneficial in the previous experiments were adopted (i.e., ConvNeXt-T pre-trained on ImageNet, despeckling applied during pre-processing, etc.). The resulting network, including the feature extraction and the detection and classification subnetworks, has a total of 39.1 million parameters. The network was trained for a maximum duration of 40 epochs for the most difficult test case (i.e., the one including all the airplane types and airports), and of 10 epochs for all the other test cases. A learning rate of $l_r = 4 \times 10^{-3}$ is used for all test cases. To monitor the training process, the AP values for the different airplane types were computed for the images in the validation set after each training epoch. Early stopping was applied to stop the training when the mean AP for the validation set began to drop. After the training was finished, the performance was evaluated by computing the AP values for the images in the test set. These were not used in the previous experiments to avoid optimizing the hyperparameter selection for the images in the test set.

In addition to this comprehensive analysis of the AP values for the different test scenarios, the detection results were also qualitatively evaluated, and the method's runtimes during both training and inference were analyzed. For the qualitative evaluation, both the detected and annotated bounding boxes were drawn over the SAR images, enabling a visual verification of the accuracy of the detections. The predicted and annotated object classes, and the confidence score for each detection were also written next to the corresponding bounding boxes.

As previously introduced, prior knowledge delimiting the possible locations of the airplanes (e.g., on aprons, taxiways, runways, etc.) is typically available, as airports tend to be accurately mapped and rarely change. For example, this information could be obtained from OSM data, using the Overpass API as previously done for the monitoring of oil storage tanks. Because the main goal here was to assess the perf This information was not used here, as the goal was to assess the performance of the proposed ATR method. However, in a real application scenario, the use of this prior knowledge could potentially eliminate some false detections and also speed up the inference time, by reducing the number of pixels to be processed.

Finally, the previously introduced CD method was also applied here in combination with the ATR method, showing that it can provide valuable complementary information. While the ATR method can detect and classify the airplanes in each image, it cannot determine whether an airplane which is parked in the same spot in two consecutive images moved or remained stationary. Therefore, by itself, the ATR method cannot provide reliable information on the number of departures and arrivals. On the other hand, the CD method can accurately detect changes due to the movement of man-made objects inside the airport and estimate the time during which of these objects remained static, but cannot determine which of these changes actually correspond

to airplanes. The information provided by these two methods can be combined to estimate the time of arrival and departure of each of the detected airplanes. Besides, this will also allow to detect activity associated to other types of objects for which no training data is available.

All these results will be shown in Section 8.3.

8 Results

As described in Chapter 7, the methods presented in this thesis were applied to real SAR data to solve specific problems involving the monitoring of different types of human activity. In this chapter, the obtained results will be shown and briefly analyzed. Section 8.1 shows the results for the monitoring of oil storage, Section 8.2 for the monitoring of construction activity, and Section 8.3 for the monitoring of airport activity. All the SAR images shown in this chapter were transformed for visualization as described in Section 3.1. Part of the material in this Chapter has been published in [Villamil Lopez & Stilla, 2021] and [Villamil Lopez & Stilla, 2022].

8.1 Monitoring of oil storage

As described in Section 7.1.3, the method proposed in Chapter 4 was applied to all the storage tanks in the dataset of the port of Rotterdam, which was introduced in Section 7.1.1. Here, the results of the different experiments done to evaluate the method's performance will be shown and analyzed. First, the estimation of the dimensions of the oil storage tanks will be evaluated both qualitatively (by performing a visual accuracy assessment) and quantitatively (by comparing the obtained values with those obtained in manual measurements). Then, the performance of the proposed classification approach to distinguish between tanks with a fixed and a floating roof will be evaluated and compared for different classifiers and different amounts of training data. Finally, a short overview of the method's runtimes will be provided.

8.1.1 Visual accuracy assessment

Initially, due to the large amount of oil storage tanks in the scene and the lack of ground truth data, visual accuracy assessment will be performed. For this, the detected semicircular double reflections are drawn over the SAR amplitude image. Only half of each detected semicircle will be drawn, as this makes it easier to determine whether it accurately matches the actual semicircular double reflection on the SAR image. The resulting images are then analyzed by visual inspection to determine the number of tanks for which all the double reflections have been correctly detected, as this will imply that all the parameters have been correctly estimated. The results of this visual accuracy assessment are summarized in Table 8.1.

In this table, results are shown separately for tanks with a fixed and a floating roof, in order to assess whether the method performs better for a specific type. Additionally, results obtained using a single image and the complete time series are compared. Both of these methods were applied twice: once without any prior information about the radius of each storage tank (i.e., radius can have any value between 10 and 50 m), and once using the approximate radius value $\hat{r}_t$ obtained from OSM to limit the possible radius values to a smaller interval given by $\hat{r}_t \pm 5$ m.

These results show that all the variants of the proposed method perform very well for storage tanks with a floating roof (which, as previously mentioned, represent the most relevant use case), with all parameters being correctly estimated for 91.66% to 96.87% of the tanks depending on

Table 8.1: Results of the visual accuracy assessment.

Method	Radius	Roof type	Correct	Wrong	Percent correct
Single image	10 – 50	Floating	88	8	91.66%
		Fixed	54	17	76.05%
Single image	$\hat{r}_t \pm 5$	Floating	90	6	93.75%
		Fixed	62	9	87.32%
Time series	10 – 50	Floating	91	5	94.79%
		Fixed	61	10	85.91%
Time series	$\hat{r}_t \pm 5$	Floating	93	3	96.87%
		Fixed	65	6	91.54%

the method variant. For tanks with a fixed roof the percentage of tanks for which all parameters are correctly estimated is lower, especially for some of the method variants, with this percentage being only 76.05% when using a single image and no prior information about the tank radius, but increasing up to 91.54% when using a time series and the approximate radius value from OSM.

The results obtained using the time series and the approximate radius value from OSM can be visualized in Fig. 8.1 for a subset of the imaged scene which contains over 50 storage tanks. In it, it can be seen how the semicircular double reflections have been correctly detected for all but one of the storage tanks. This tank has a floating roof and is located towards the bottom left of the image, and the double reflection at its bottom (shown in blue) was wrongly detected at a false position. This implies that the maximum capacity and the amount of oil stored in that particular tank will be wrongly estimated. However, the changes in the amount of oil storage will be correctly estimated anyway, as the radius and the vertical positions of the floating roof were correctly estimated. While Fig. 8.1 only shows the results obtained for the first image in the time series, the vertical location of the floating roofs was correctly detected for all the storage tanks in all three images.

8.1.2 Quantitative accuracy analysis

After the qualitative evaluation of the obtained results via visual analysis, a more detailed analysis of the accuracy achieved will be performed for 10 of the storage tanks shown in Fig. 8.1 (those enclosed by white rectangles and labelled with the corresponding numbers). Tanks 1 to 5 have a floating roof, whereas tanks 6 to 10 have a fixed roof, and all of them have different sizes. As no ground truth data is available regarding the dimensions of these storage tanks or the amount of oil stored in them, this accuracy analysis will be performed by comparing the dimensions automatically estimated by the proposed methods (using a single image and a time series) with those obtained from manual measurements. These manual measurements will be performed as described in [Hammer et al., 2017], as this measurement principle has been validated. For this comparison the use of approximate radius information from OSM will not play any role, as it only reduces the number of tanks for which the method does not work properly by setting a tighter limit for the possible values of a tank's radius, and it does not have any effect in the accuracy of the estimated values for the cases in which the method works properly.

Table 8.2 shows the comparison for the tank radius r_t and height h_t (both given in meters). Here, small differences can be seen between the results obtained from a single image and those obtained from the time series. Additionally, the results obtained from these two methods differ slightly from those obtained from manual measurements, which are performed simply by selecting

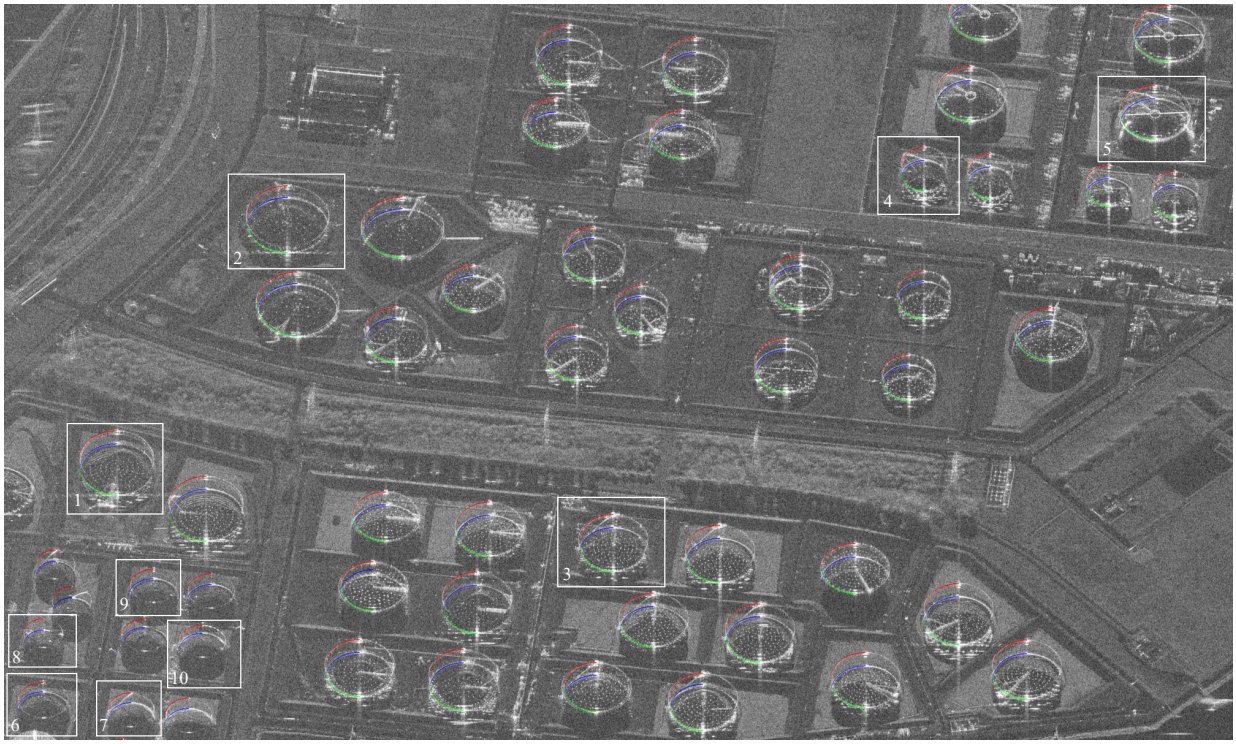


Figure 8.1: Visualization of the results for many oil tanks of different sizes for a single date, demonstrating the method's performance. Only half of each semicircle is drawn to make the comparison to the SAR amplitude image easier. The ten highlighted tanks will be used for a quantitative accuracy analysis.

Table 8.2: Results for the estimation of the tank size.

Tank	Single image		Time series		Manual measurement	
	r_t	h_t	r_t	h_t	r_t	h_t
1	39.71	21.11	40.96	21.11	40.73	21.79
2	44.33	19.07	44.33	19.07	45.36	18.84
3	37.45	21.79	36.95	21.79	38.03	22.13
4	25.32	22.48	24.83	22.48	25.16	23.04
5	38.28	21.79	37.78	21.79	38.26	22.02
6	29.20	14.98	28.96	14.98	27.33	15.32
7	25.23	16.35	25.23	16.35	25.25	17.03
8	20.44	19.07	21.19	18.39	21.23	18.73
9	27.48	14.98	27.73	15.66	25.36	17.93
10	25.48	12.94	25.48	12.94	25.26	12.26

a couple of image points, and therefore they are not necessarily more accurate. Except for some small differences, all the estimations from the different methods seem to agree. To put these differences into perspective, an error of one pixel will translate into an error of 0.68 m in the estimation of h_t, and of 0.32 m (which is the corresponding square pixel size on the ground) in r_t. Therefore, most of these differences between the different methods are in the order of one to a couple of pixels, with these differences being often slightly larger for the estimated radius than for the height.

Table 8.3: Results for the estimation of the floating roof height.

Tank	Single image	Time series			Manual measurement		
	h_{r_1}	h_{r_1}	h_{r_2}	h_{r_3}	h_{r_1}	h_{r_2}	h_{r_3}
1	4.09	6.81	6.81	6.81	6.83	6.83	6.83
2	8.17	8.17	17.70	15.66	10.09	19.63	17.47
3	12.26	10.90	10.90	23.16	13.31	13.31	25.57
4	6.13	5.45	14.98	14.98	6.51	16.16	16.16
5	17.03	17.71	17.71	21.11	18.91	18.91	22.55

In addition to the radius and height of each tank, for the tanks 1 to 5 the vertical position of its floating roof at the time of each image acquisition must be estimated. When applying the method to a single image (the first one in the series) only the initial height of the floating roof h_{r_1} was estimated. When applying the method to the entire time series, and in the manual measurements performed for comparison, the height of the floating roof was estimated for all three images. The obtained results can be seen in Table 8.3. Again, the results obtained from the manual measurements are not necessarily more accurate, but they serve as a useful reference for this comparison in the absence of ground truth data. In this table, some significant differences can be observed across the different estimates for the initial height of the floating roofs. On the other hand, the vertical displacements of the floating roof in between consecutive images (i.e., the difference between $h_{r_{i+1}}$ and h_{r_i}) estimated with the proposed method agree very well with the manual measurements. The main cause of these differences in the initial height is actually the difference in the previously estimated radius r_t (shown in Table 8.2). For the used incidence angle, an error of 1 m in r_t will induce an error of 2.23 m in h_{r_1}. This is caused by the fact that the semicircular reflection of the floating roof appears towards far-range, whereas the two reflections at the outer tank structure appear toward near-range. Therefore, this is not a particular issue of the proposed method, but it will be common to all the methods employing SAR images.

8.1.3 Classification performance

For each of the storage tanks in this dataset, the corresponding features were computed from both a single image and the time series, as described in Chapters 4.2.4 and 4.3.5, respectively. The classification results using these two different sets of features (obtained from a single image and a time series) were compared for different amounts of training data and two different classifiers: a support vector machine (SVM) and a random forest. To purely evaluate the classification without the influence of the errors occurred during the previous steps, the few tanks for which the estimation of the tank dimensions did not work properly (i.e., those listed in Table 8.1 as wrong) were removed from the dataset.

In order to analyze the effect that the amount of training data has in the classification results, both the SVM and random forest classifiers were trained multiple times from scratch using training datasets of different sizes (with 10, 25, 50, 75 and 100 samples). Each time the training samples were selected randomly while keeping an equal split of tanks with floating and fixed roofs, and the remainder of the dataset was used as test data. To minimize the influence that the selection of the training samples has in the results, this process was repeated 100 times for each case, and the F_1 score of the corresponding 100 trained classifiers was averaged, obtaining an average F_1 score for each number of training samples. The results obtained for both classifiers and both sets of input features (obtained from a single image and from the time series) can be seen in Fig. 8.2. The best classification results are consistently obtained for all the different amounts of training data

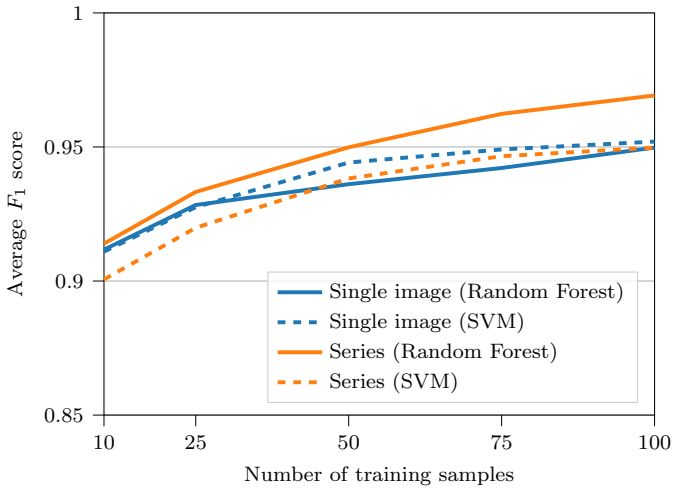


Figure 8.2: Comparison of the classification performance using the features extracted from a single image and a time series for two different classifiers and different amounts of training data.

when using the input features obtained from the time series and a random forest classifier, which reaches an average F_1 score of 0.97 with 100 training samples. When using the input features obtained from a single image, the SVM classifier performs slightly better than the random forest, reaching an average F_1 score of 0.95 with 100 training samples.

8.1.4 Runtime analysis

The proposed method could potentially be applied to monitor the amount of oil stored in many oil refineries, each with a large number of storage tanks, regularly every few days. Here a brief analysis of the runtime of the proposed method will be provided, to give an outlook on how the processing time scales up with the number of SAR images and/or number of storage tanks to be monitored. This processing time will be analyzed separately for two separate stages of the proposed method: pre-processing (which involves the detection of CSs in the SAR images, as well as change detection when using a time series), and information extraction (which involves the automatic estimation of all the relevant parameters for each storage tank).

For the pre-processing stage the runtime will depend on the number of SAR images and their sizes. When using a single SAR image the pre-processing involves just the CS detection which, for one of the used TerraSAR-X images with 9014×18847 pixels, takes 2 minutes and 49 seconds on an Intel i7-8665U laptop CPU (with 4 cores) and 32 GB of RAM. When using a time series the pre-processing also involves co-registration and coherent change detection in addition to the CS detection. For the series of three TerraSAR-X images used here, this takes a total of 10 minutes and 48 seconds (or 3 minutes and 36 seconds per image) on that same hardware. In both cases, the pre-processing algorithm was implemented in Python, using Numpy compiled with Intel MKL.

Regarding the information extraction stage, its runtime will mainly depend on the number of storage tanks in the imaged scene, as the processing will be performed separately for each storage tank using just a small image patch around it. As with the pre-processing stage, the code was implemented in Python using Numpy, and it was tested on the same hardware. The total runtimes, as well as the average time per storage tank, are listed in Table 8.4 for the different methods (using a single image or a time series, with or without prior knowledge of the approximate radius). The runtimes for the methods processing the time series represent the time required for processing all three images in the series (i.e., including the estimation of the displacements of the floating roofs), and not the time per image.

Table 8.4: Results of the runtime analysis for the information extraction stage.

Method	Radius	Total runtime	Seconds per oil tank
Single image	10 – 50	27 min 18 sec	9.80
Single image	$\hat{r}_t \pm 5$	2 min 54 sec	1.04
Time series	10 – 50	19 min 50 sec	6.12
Time series	$\hat{r}_t \pm 5$	2 min 30 sec	0.89

8.2 Monitoring of construction activity

As described in Section 7.2.3, the CD method proposed in Chapter 5 was applied to monitor construction activity in the city of Munich. For this, a time series of 49 images acquired over a period of almost 3 years was used, which was introduced in Section 7.2.1. Here, the obtained results for the detection of changes due to the construction of new buildings and renovations to existing ones are verified by visualizing sequences of SAR and/or optical images. Then, the results demonstrating the detection of other types of changes, with examples for the build-up of different festivals taking place in the same park, are shown and briefly analyzed. Finally, some results illustrating the method's capability to segment static and unchanged man-made objects are also shown, and a short overview of the method's runtime is provided.

8.2.1 Detection of new and renovated buildings

The obtained results for an area around the city campus of the Technical University of Munich (TUM) are shown in Fig. 8.3a. This image shows the latest SAR image in the series (acquired on February 28, 2019), with the segmented changes highlighted in different colors according to the date when the construction work finished. Visual inspection of the SAR time series allows to validate the obtained results, as it can be seen that renovation works have actually taken place at all the highlighted locations. In some cases, these renovations resulted in visible alterations to the corresponding buildings, while in others they could only be identified by the temporary presence of scaffolds. Many other changes occurred in this area and were also detected by the proposed method, but all those not fitting the desired temporal pattern were discarded. This successfully discarded changes corresponding to moving objects and other activity. However, it also discarded two changes due to the construction of two new buildings which were still not completely finished by the acquisition time of the last image.

Further verification using optical images has been performed for the museum "Alte Pinakothek" and for one side of the TUM building, both areas enclosed in Fig. 8.3a by white rectangles. The renovation process of the "Alte Pinakothek" can be seen in three optical images at the bottom of Fig. 8.3 (l-m) and the five SAR images in the second row of Fig. 8.3 (b-f), each with their corresponding acquisition date. In the same way, the changes at the TUM building can be seen in an optical image and a street level photography in Fig. 8.3o and 8.3p, respectively, as well as in the five SAR images in the third row of 8.3 (g-k). In both cases the results shown in Fig. 8.3a agree with the sequence of the renovation process depicted in both the optical and SAR images, where proof of these changes was manually outlined using the same colors. The optical images show all the renovation works at both buildings, except for the one highlighted in green on the university building in Fig. 8.3a, as no optical images acquired at that time could be found. The SAR image sequences do however show all the detected changes, including that one: the scaffold can be seen in Fig. 8.3g and no longer at 8.3h (in both cases highlighted in green). In some cases the segmented changes shown in Fig. 8.3a are smaller than the manually outlined areas in the SAR and optical images. The reason for this is that the results shown in this figure only

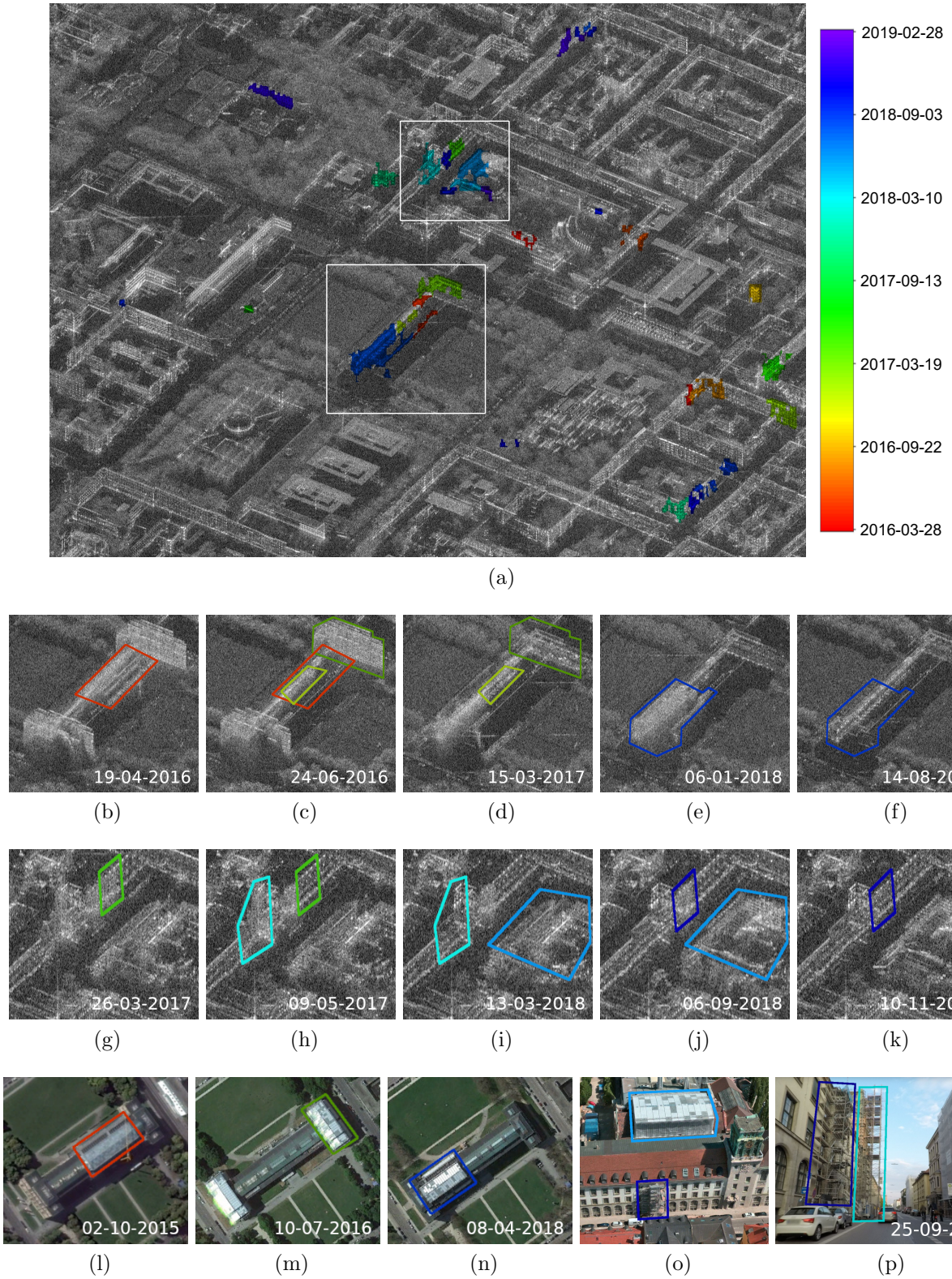


Figure 8.3: Results for the detection of changes due to the construction and renovation of buildings in an area around the city campus of the TUM. For visual verification, some optical images and a few of the SAR images in the series are shown for two buildings: the museum "Alte Pinakothek" and a side of the university building. a) Segmented changes drawn over the last SAR image of the series, with the color denoting the date at which the construction work was finished, b-f) Sequence of SAR images for the museum, g-k) Sequence of SAR images for the university. l-n) Optical images for the museum, o-p) Optical images for the university. The optical satellite images were obtained from Google Earth (l-n) and Apple Maps (o), and the street level photography from Mapillary (p). The acquisition date of the optical image in (o) was not listed in Apple Maps, but it appears to have been acquired in mid 2018.

highlight the parts of the final building structures which were modified, and not all the image pixels which changed at some point throughout the time series (e.g., those temporarily covered by scaffolds). Also, the renovation of the building on the right side of the street in Fig. 8.3p could not be detected, as this façade is not visible to the SAR sensor and appears as a shadow area in the SAR images. This change could also be detected by applying the proposed method to a time series acquired with descending orbit.

In addition to the results shown for the TUM area, many more changes due to the construction or renovation of buildings were detected at different locations across the city. Figure 8.4 shows four of the detected changes, each displayed in a different row (rows a to d). Each change is illustrated by three SAR image chips acquired before (column A), during (column B) and after (column C) the construction work. The detected changes are highlighted in the final images (i.e., Fig. 8.4C), with the colors corresponding to the date on which the construction work finished, using the same legend from Fig. 8.3a. Fig. 8.4a and 8.4b show changes due to the construction of new buildings, whereas Fig. 8.4c and 8.4d show examples due to renovations. Some of these changes appear in multiple colors, meaning that different structures were finished at different times (like in Fig. 8.4b), or that different sections of the scaffold were gradually removed (like in Fig. 8.4d). Again, visual inspection of the SAR images in the time series allowed to validate the accuracy of the obtained results. For the two new buildings, the image sequence shows an empty lot, followed by the construction progressing and finally the finished building. On the other hand, for the renovations the first and final states are very similar, with the presence of a scaffold during the renovation being the main change. Even if these two types of changes are different, the proposed method cannot distinguish them as they exhibit similar temporal patterns (i.e., the appearance of new CSs that later remain static).

One possible way to distinguish changes due to the construction of new buildings and the renovation of existing ones would be to evaluate the number of CSs over time inside the segmented changes. It has been shown that a significant increase in the number of CS indicates an increase in the amount of built-up structures [Villamil Lopez & Stilla, 2019]. This information is plotted in Fig. 8.5 for the four changes shown in Fig. 8.4 (a to d). As expected, for the examples corresponding to the construction of new buildings (Fig. 8.4a and 8.4b) the number of CSs increases by a significant amount as the construction progresses and remains constant after it finishes. For the renovations (Fig. 8.4c and 8.4d) there is only a small variation in the number of CSs between the prior and final states, with a larger variation during the time where the scaffold is present. This kind of analysis could potentially be used to achieve a finer classification of the detected changes. However, this task is out of the scope of this work.

8.2.2 Detection of other changes

In addition to the construction and renovation of buildings and infrastructure, the proposed method also detects other changes like the build-up for festivals and/or any events taking place, the movement of objects, etc. As mentioned in Section 5.3.3, such changes typically involve CSs both appearing and disappearing inside relatively short periods. To show this, the proposed approach was applied to detect only changed objects which remain unchanged and static for less than an interval ΔT of 4 months. Some of the results are shown in Fig. 8.6 for a small area around the park "Theresienwiese" where different festivals take place throughout the year. Figure 8.6a through 8.6c show this area at three different times along with the corresponding changes. The detected changes were highlighted in a color ranging from red to yellow (with the hue component varying linearly from 0 to 60) according to the length of time that each object is present and remains unchanged. Most of these changes appear in dark red (hue value of 0), meaning the corresponding objects were only present during one image. A few objects are highlighted in

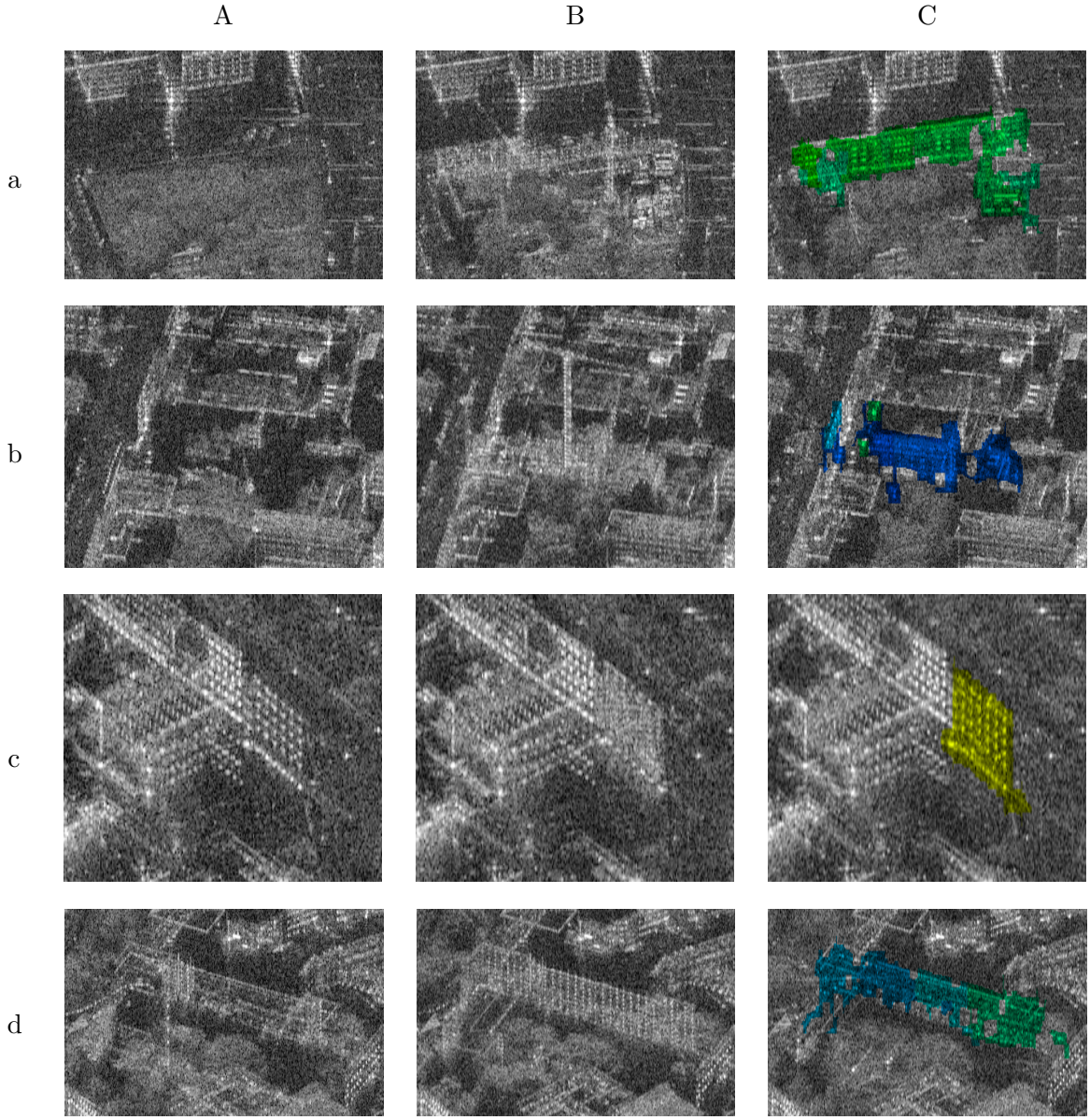


Figure 8.4: Four examples (rows a-d) of detected changes to buildings. The three columns correspond to three images showing each change: A) images acquired before the construction work, B) images acquired during the construction work, C) final images with the detected new and/or modified structures highlighted.

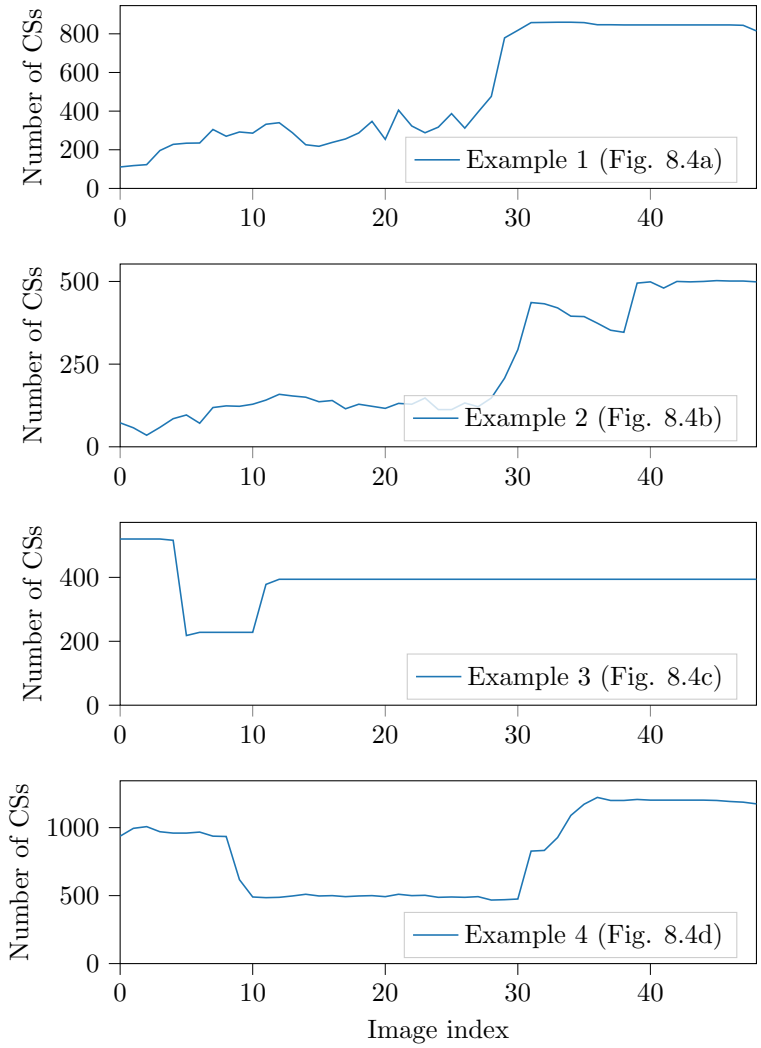
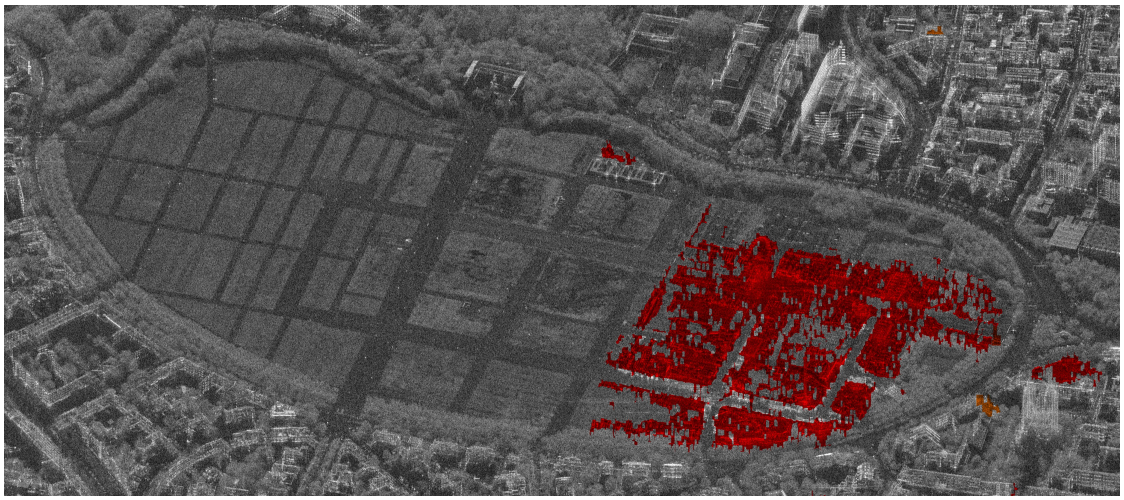


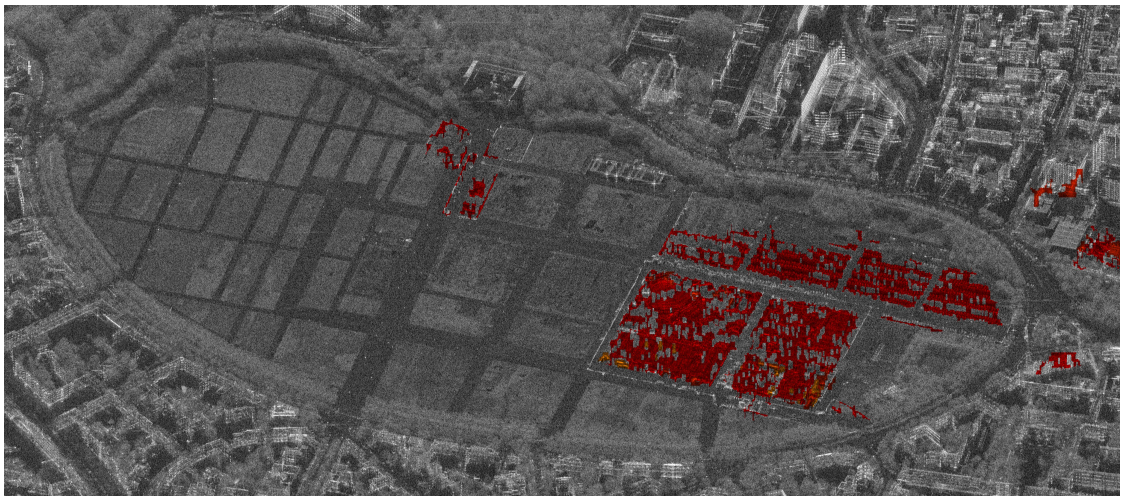
Figure 8.5: Evolution of the number of CSs over time for the four changes in Fig. 8.4, with those corresponding to the construction of new buildings (Fig. 8.4a and 8.4b) and renovations (Fig. 8.4c and 8.4d) showing distinct patterns.

orange, such as the roofs of some of the large festival tents in Fig. 8.6c, showing that these structures were built before and/or stayed longer in the scene than the other objects.

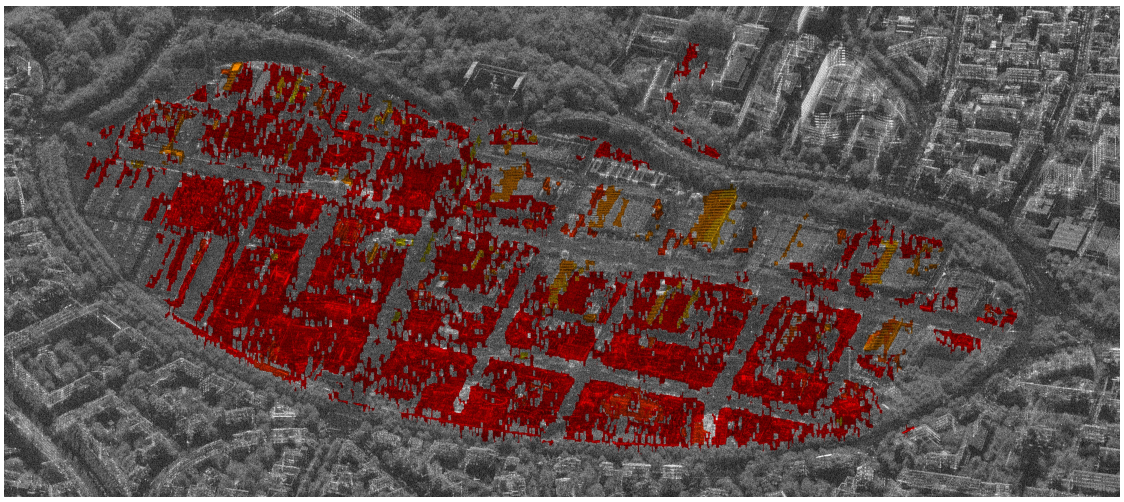
In addition to the changes due to the festivals in this park, changes were also detected for some buildings where construction work is taking place. The analysis described in the previous subsection allowed to detect the final change to buildings and estimate when the construction or renovation finished. However, during the construction phase the buildings change continuously, and all these fast changes are also detected. Once the final change to a building is detected, all the previous changes in the same overlapping area could be identified as the corresponding construction work.



(a)



(b)



(c)

Figure 8.6: Detected changes at the park "Theresienwiese" due to objects both appearing and disappearing within a time period of less than 4 months, highlighted in a color according to the length of time that each object is present. The three figures show this park at three different dates, with different events taking place which cause the majority of the detected changes: a) April 19, 2016; b) December 15, 2017; c) September 27, 2018.

8.2.3 Detection of unchanged and static objects

The proposed method also detects the man-made objects that remain static and unchanged throughout the time series. For an urban scene like the one in this dataset, these correspond to the existing buildings and infrastructure where no renovations took place. Figure 8.7 shows an example of the obtained results for an area close to the Munich city center. The unchanged CSs are highlighted in blue over the first and last images of the time series in Fig. 8.7a and 8.7b, respectively. These CSs are similar to the persistent scatterers of PSI methods [Ferretti et al., 2001], but can be obtained with as few as two images. However, especially in layover areas (e.g., the building façades), the resolution loss in range due to the sublooking required for the CS detection results in less point scatterers being detected than when using PSI. The proposed object-based analysis method was also applied to segment the unchanged objects. The segmentation results can be seen in Fig. 8.7c and 8.7d, again highlighted over the first and last images of the series, respectively. Comparison of the left and right columns (corresponding to the first and last images) of Fig. 8.7 show how the proposed method correctly identifies unchanged objects, consisting mostly on buildings, but also street lamps, etc. For some of the buildings which are not highlighted the changes can be clearly seen (e.g., one building towards the top, and another one towards the bottom left). For others, no changes can be seen between the first and last images, but visual inspection of the complete time series revealed that renovations took place.

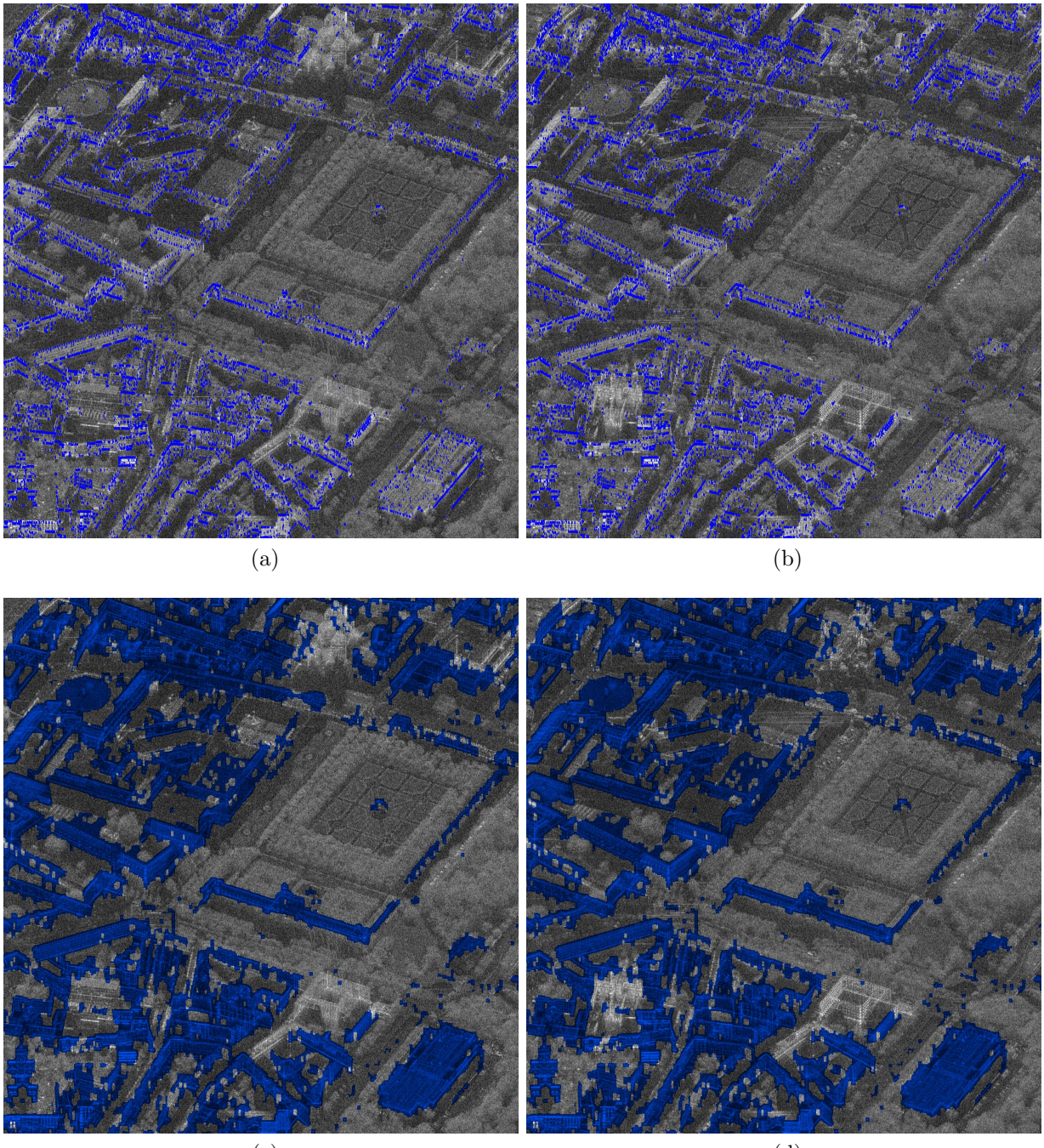
8.2.4 Runtime analysis

Here a brief analysis of the runtime of the proposed method will be provided, to give an outlook on how the processing time scales up with the number of SAR images. This processing time will be analyzed separately for two different stages of the proposed method: pre-processing (which involves the co-registration of the SLC images, CS detection, and calculation of the CD metric), and the actual change detection (which involves evaluating the CD metric for the pixels containing CSs, clustering them, and segmenting the detected changes).

The duration of the pre-processing stage will depend on the number of SAR images and their size. In this case, the time series contained 49 images, and the one used as a master for the co-registration had a size of 7816×18337 pixels. The pre-processing, implemented in Python using Numpy compiled with Intel MKL, took approximately 2 hours and 5 minutes (or 2 minutes and 33 seconds per image) on an Intel i5-1135G7U laptop CPU (with 4 cores) and 24 GB of RAM. Whenever a new image is added to an existing time series, the pre-processing does not need to be repeated for the complete time series. Instead, the new image can be co-registered and added to the existing image stack, and the CD metric for the last few images can be updated.

The change detection stage is much faster, thanks to its simplicity and the fact that most of the processing is only performed for the pixels containing CSs. Its runtime will depend on the number of changes to be segmented and their sizes. The initial step of evaluating the CD metric for the pixels containing CSs is very fast, but the subsequent clustering and segmentation steps are more time consuming. However, these two steps can be applied selectively, only for those CSs with a certain temporal behavior. For example, the detection and segmentation of the changes associated to newly constructed and renovated buildings took just 43 seconds for the complete time series. As with the pre-processing stage, this code was implemented in Python using Numpy, and it was tested on the same hardware.

When attempting to segment changes corresponding to objects that are present in only one image, a despeckling method also needs to be applied. The time required for despeckling was not included in the previous analysis, as this is an optional step, and its runtime will also vary



(a) (b)

(c) (d)

Figure 8.7: Objects that remained unchanged throughout the time series. Top: detected unchanged CSs highlighted in blue over: a) the first image, b) the last image. Bottom: segmentation results for the unchanged objects highlighted in blue over: c) the first image, d) the last image.

significantly depending on the chosen despeckling method. Nevertheless, despeckling does not need to be applied to all the images in the series, as it can be selectively applied during the change segmentation step, only for small image patches and for those cases where it is required.

8.3 Monitoring of airport activity

As described in Section 7.3.3, the ATR method proposed in Chapter 6 was applied to detect and classify the airplanes present in TerraSAR-X images of different airports, using the dataset introduced in Section 7.3.1. Here, the obtained results will be presented. First, the precision-recall curves and the corresponding average precision (AP) scores obtained for the different test cases will be shown and analyzed. A detailed description of these metrics can be found in [Padilla et al., 2020]. After this analysis, the detection results will be shown and compared with the annotations for several example image patches. Additionally, a short overview of the method's runtimes both during training and inference will be provided. Finally, some example results will be shown to illustrate how the previously introduced CD method can be applied here to estimate the time of arrival and departure of each of the detected airplanes.

8.3.1 Precision-recall curves and average precision for the different test cases

Here, the results for all the four test cases will be analyzed. Because different splits have been proposed to divide the data of each test case into training, validation and testing, the network will be trained and evaluated multiple times for each test case. The results obtained for the different splits of each test case will be compared to gain some valuable insights on the generalization capabilities of the proposed method. This analysis will show how much the performance drops when the test images are acquired under conditions which are not available in the training images.

The results will be analyzed using the precision-recall curves computed for the test images, as these illustrate the detection performance for all the possible similarity thresholds. The precision-recall curves will be computed here using 100 points (varying the detection threshold from 0.01 to 1 in steps of 0.01), and only those detections with an IoU above 0.5 will be considered correct. For comparison purposes, the precision-recall curves for the images in the validation set will also be shown. For the test cases with several airplane types, these curves will be shown separately for each type to illustrate how the detection performance varies across the different object classes.

In addition to the precision-recall curves, the AP scores for the test images will also be computed and listed. These AP scores represent the area under the corresponding precision-recall curves, and will also be computed separately for the different airplane types. Additionally, a mean AP will also be computed by averaging the AP scores over all the available classes. The AP scores obtained for the different splits of all the test cases are listed at the end in Table 8.5. The number of samples used for training, validation and testing in all these splits were previously listed in Section 7.3.1 when the dataset was introduced. However, because these numbers are important for the analysis of the obtained AP scores, they will be listed here once again in Table 8.6 for convenience.

Results for the first test case: one airplane class in a single airport

The precision-recall curves for the three different splits of the first test case can be seen in Fig. 8.8. The number of samples used for training, validation and testing in the different splits are listed above the corresponding plots. The corresponding AP values (for the test images) are listed in the top of Table 8.5. The results are very good for all the three different splits.

For the "varied split", which represents the ideal case with similar conditions for training and test, an almost perfect AP score of 0.993 was obtained. The precision-recall curve, shown in Fig. 8.8a, reveals that for a certain similarity threshold all the airplanes are correctly detected (recall of 1), but with a few false positives (precision of 0.959). The precision-recall curve for the validation set shows that the performance for the validation images is even slightly better.

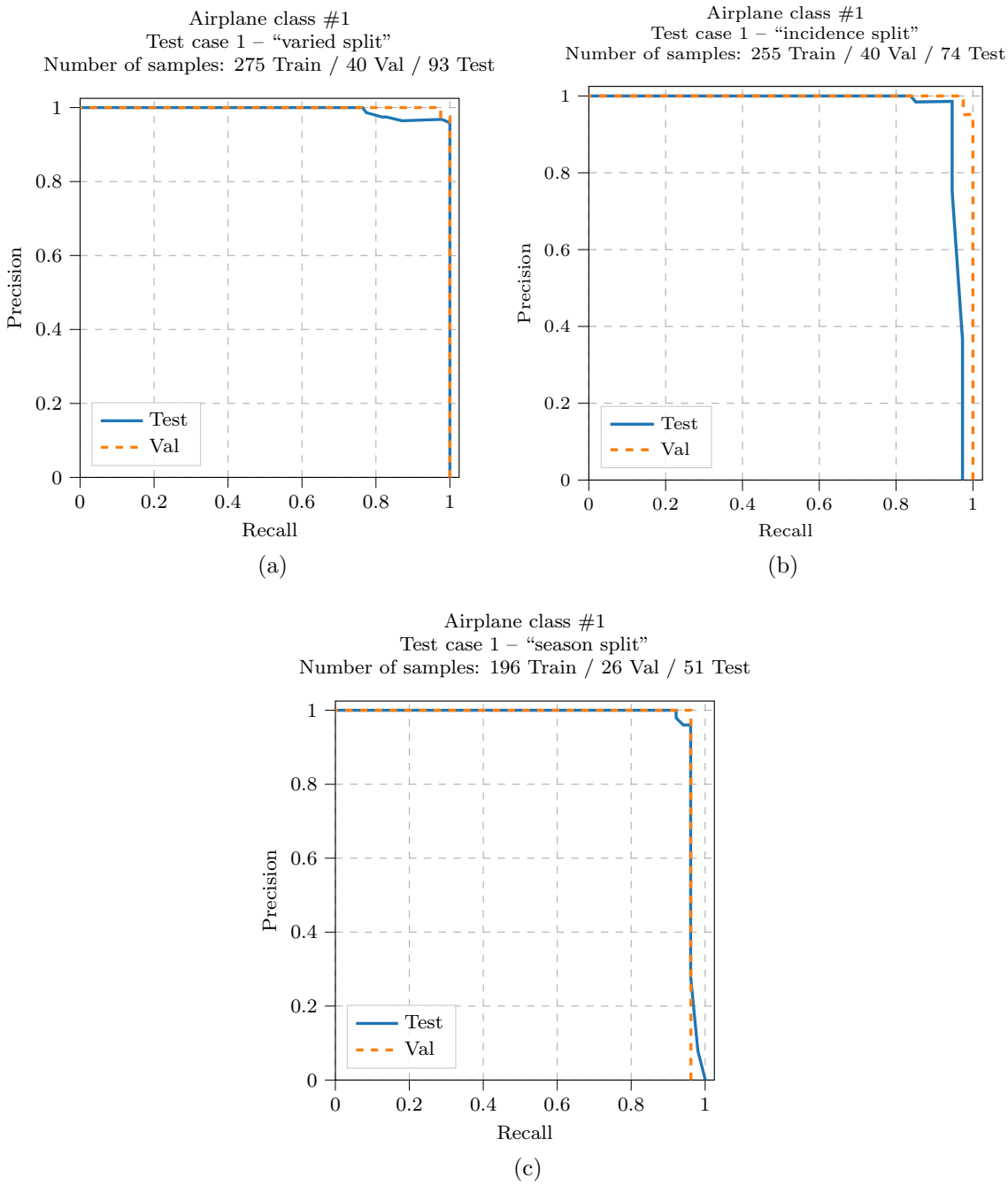


Figure 8.8: Precision-recall curves for the first test case (one airplane type in one airport). Results for the different splits: a) "varied split", b) "incidence split", c) "season split".

The results of the "incidence split" show that, as expected, the method's performance will drop when applied to images with incidence angles significantly different from those available in the training set. Nevertheless, a very good AP score of 0.960 could still be achieved for this split. Compared to the AP score of 0.993 obtained for the "varied split", the performance drop is not that significant. The precision-recall curves for the test and validation images can be seen in Fig. 8.8b. The curve for the test images shows that for some thresholds very good precision and recall values can be achieved, although no threshold results in the detection of all aircrafts. A better performance was achieved for the validation images, which is to be expected, as these have similar conditions to the images used for training.

Finally, the "season split" illustrates the impact of seasonal effects (in this case snow) which are not present in the images of the training set. In this case, the performance drops even less, with an obtained AP score of 0.964. The precision-recall curves for this split can be seen in Fig. 8.8c. The curve for the test images shows that here the recall is slightly higher than in the "incidence split". Interestingly, here a similar detection performance is achieved for the validation and test images, even though the validation images should be more similar to those used for training.

Results for the second test case: one airplane class in five airports

For the second test case, the complexity was slightly increased by including different locations (i.e., more airports). The number of available TerraSAR-X images and airplane samples also increased accordingly. The precision-recall curves for the two different splits of this test case can be seen in Fig. 8.9. The corresponding AP values for the test images can be found in Table 8.5. Again, the results are very good for both splits.

The precision-recall curve for the "varied split" can be seen in Fig. 8.9a. This curve and the associated AP score of 0.998 indicate that a near perfect detection performance was achieved. These results suggest that adding more locations to the dataset does not affect performance as long as the different locations are available in the training data. On the other hand, when the method

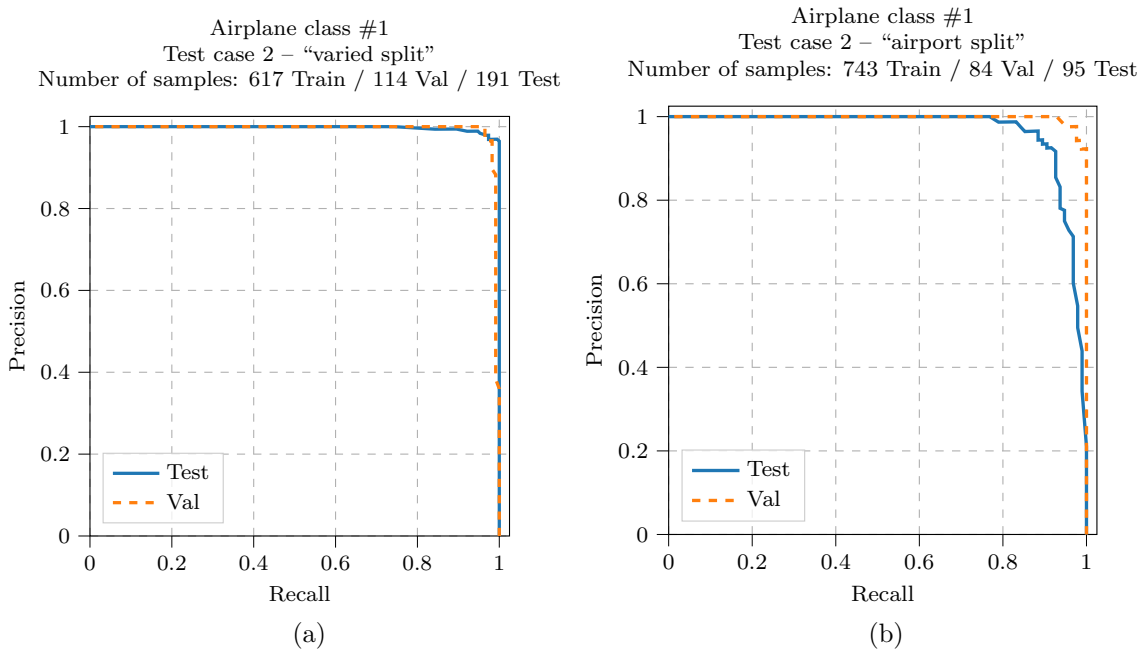


Figure 8.9: Precision-recall curves for the second test case (one airplane type in all airports). Results for the different splits: a) "varied split", b) "airport split".

is applied to locations which are not available on the training set, the detection performance will become slightly worse. This can be seen in the precision-recall curve for the "airport split", shown in Fig. 8.9b. Nevertheless, the AP score of 0.967 for this split shows that the method still performs very well in this case.

Results for the third test case: four airplane classes in a single airport

Rather than including several locations, for this third test case the complexity was increased by including all four different airplane types. The number of TerraSAR-X images stayed the same as in the first test case (as only one airport is used here too), but the number of available samples increased because of the different airplane types. However, the number of samples varies significantly across the four airplane types. This needs to be considered when analyzing the results, as the results obtained using a low number of test samples are less reliable. In this case, four precision-recall curves (one per airplane type) were generated for each of the three different splits. The four curves for the "varied split" can be seen in Fig. 8.10, those for the "incidence split" in Fig. 8.11, and those for the "season split" in Fig. 8.12. All the corresponding AP values can be found in Table 8.5. The obtained results are mostly good for all three splits, with some exceptions for airplane type #3 which will be explained below when analyzing the results of each split.

For the "varied split", an almost perfect detection performance was achieved for three out of the four airplane types: AP scores of 0.995, 0.999 and 1.0 were obtained for airplanes #1, #2 and #4, respectively. However, for class #3 a low AP score of 0.111 was obtained. The precision-recall curve in Fig. 8.10c reveals that 8 out of the 9 samples in the test images are never correctly detected, and the same occurs for the only sample on the validation set. An analysis of the imaging geometries for airplane #3 in this split reveals that in 7 of the 9 test samples and in the only available validation sample, the airplanes have orientations which are significantly different from those available in the training set. This explains why most of the instances of airplane #3 cannot be properly detected. Even for very low similarity thresholds, the recall never increases, as these samples are most likely misclassified (i.e., the network considers that a template of a different airplane type has a higher similarity). Nevertheless, these results do not necessarily mean that the performance will always be that bad for airplane type #3: the reduced number of test samples makes it impossible to assess this properly. The same applies for airplane type #4, for which a perfect AP score of 1.0 was obtained, but using only the 7 available test samples.

In both the "incidence split" and "season split", the performance is only slightly worse for the classes #1, #2 and #4. This small performance drop fits the expected behavior, as seen in the previous test cases. For airplane type #3, however, the results are better than those obtained in the "varied split", and they vary significantly. For the "incidence split", with only two test samples, an AP score of 1.0 was obtained. For the "season split" an AP of 0.5 was obtained, also with two test samples. This again illustrates that the AP results obtained using few test samples are not reliable.

The AP scores for airplane type #1 in all three splits are nearly identical to the ones obtained for the first test case (using the exact same images but only one airplane class). This seems to indicate that including additional classes does not affect the performance, and training the network to detect only a specific object class does not have any advantage.

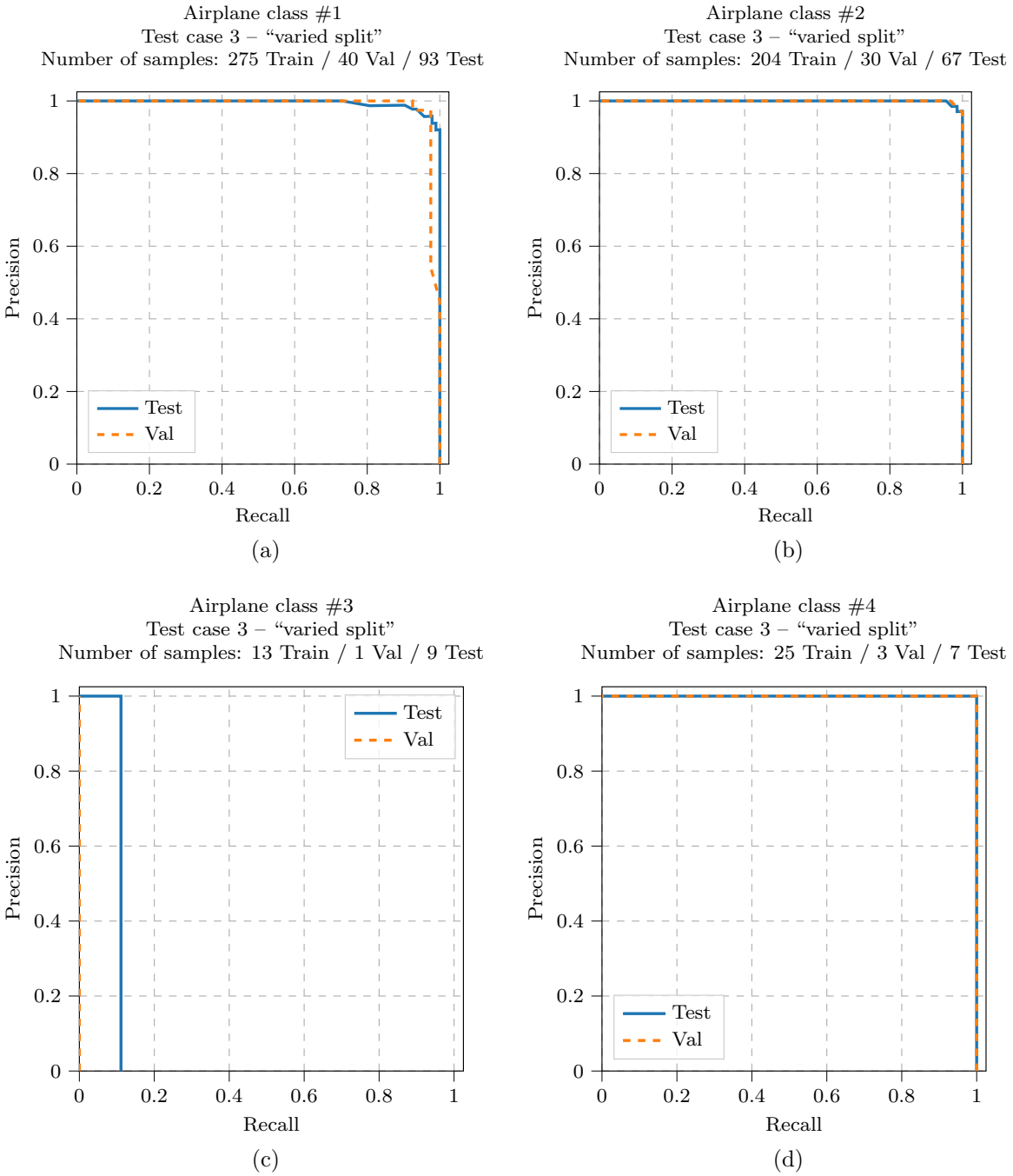


Figure 8.10: Precision-recall curves for the "varied split" of the third test case (all airplane types in one airport). Results for the different airplane types: a) #1, b) #2, c) #3, d) #4.

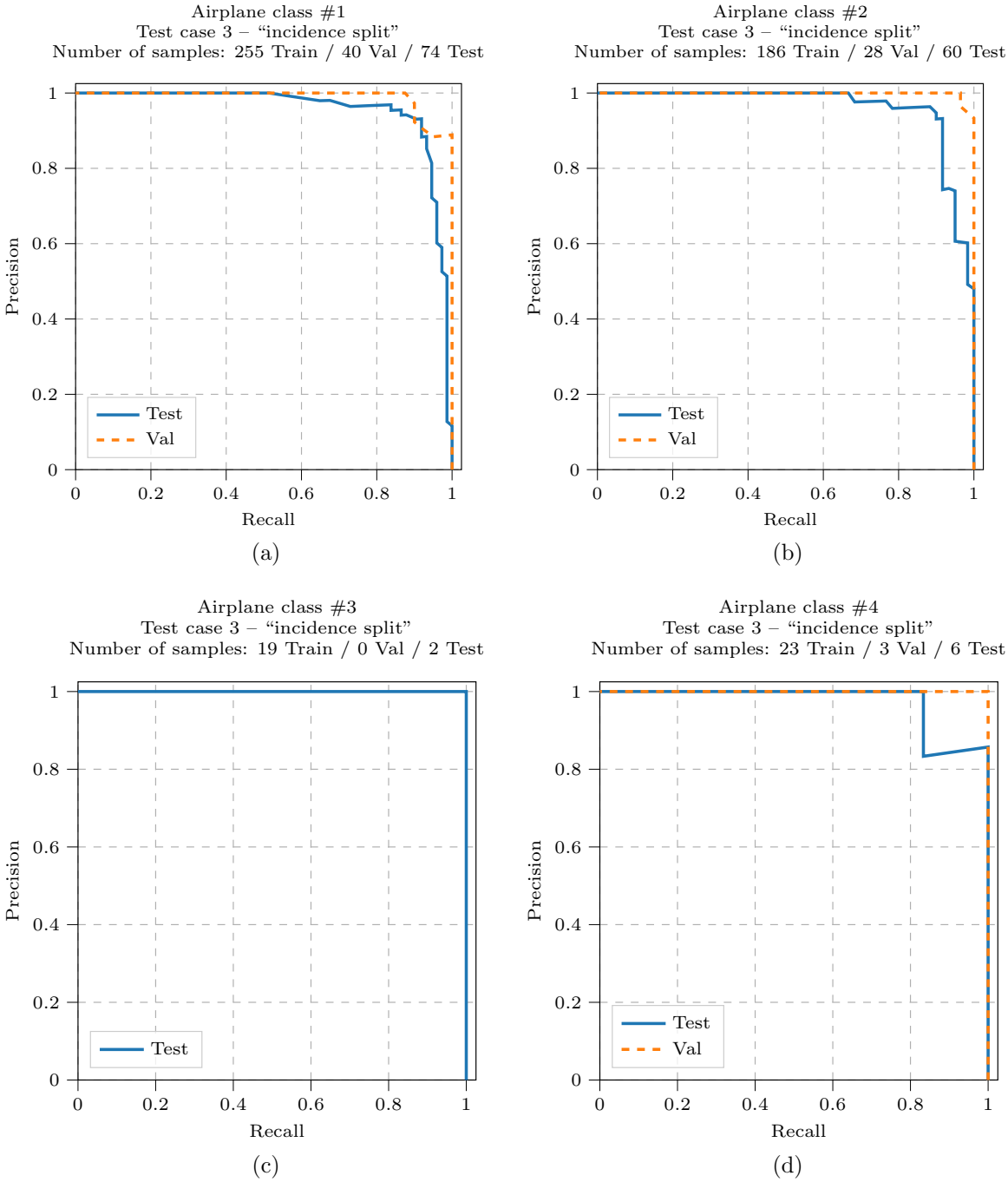


Figure 8.11: Precision-recall curves for the "incidence split" of the third test case (all airplane types in one airport). Results for the different airplane types: a) #1, b) #2, c) #3, d) #4.

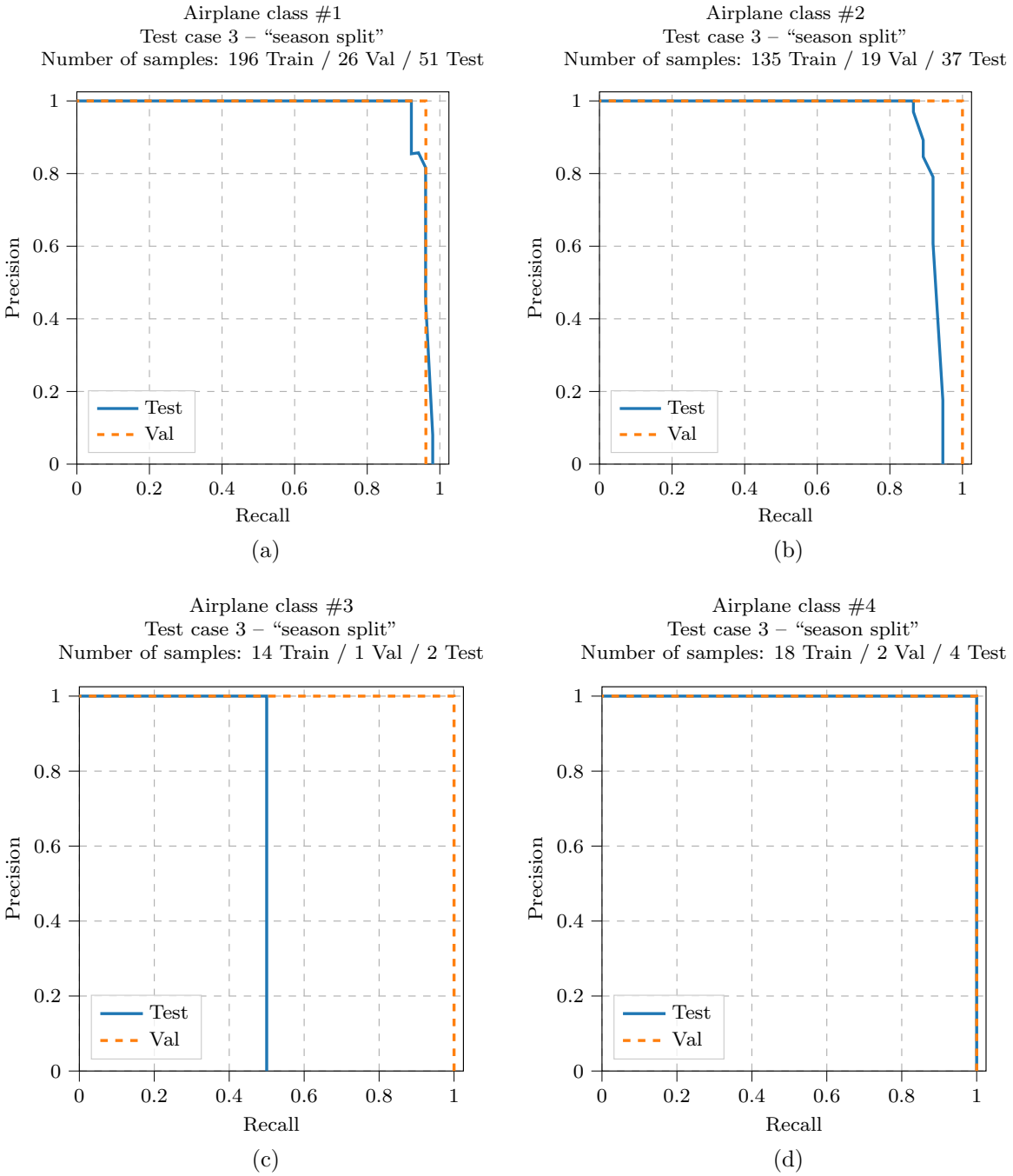


Figure 8.12: Precision-recall curves for the "season split" of the third test case (all airplane types in one airport). Results for the different airplane types: a) #1, b) #2, c) #3, d) #4.

Results for the fourth test case: four airplane classes in five airports

The last test case is also the most complex, as it includes all four airplane types in all five airports. It includes the same TerraSAR-X images as the second test case, but the number of available samples is higher due to the four different classes. The four precision-recall curves for the "varied split" can be seen in Fig. 8.13, and those for the "airport split" in Fig. 8.14. All the corresponding AP values can be found in Table 8.5.

In the "varied split", a near perfect detection performance was achieved for airplane types #1 and #2 (with AP scores of 0.998 in both cases), and a very good performance was obtained for types #3 and #4 (with AP scores of 0.866 and 0.878, respectively). This results in a mean AP of 0.935. In this case, the higher number of samples for the airplane types #3 and #4 makes the performance evaluation for these classes more reliable than in previous test cases. The lower AP scores for these classes suggests that recognizing these types of airplanes is more challenging. This is also supported by the fact that the precision-recall curves for the images in the validation set show a similar behavior. However, this might also be influenced by the lower number of training samples available for these two classes.

For the "airport split", the performance varies greatly for the different airplane types. For airplane type #1, a very good performance was achieved, with an AP value of 0.936. Interestingly, this performance dropped slightly with respect to the second test case, which included exactly the same samples for this airplane class, but without any additional classes. The performance for airplane class #2 cannot be evaluated due to the absence of test samples, and the AP value of 1.0 obtained for airplane class #3 is not reliable, as only one test sample is available. Finally, for airplane class #4 the performance is extremely bad, as indicated by the precision-recall curve, shown Fig. 8.14d and the associated AP score of 0.043. This cannot be explained by a lack of test samples, as there are 42 test samples for this class, which implies that the method consistently performs bad at detecting all the instances of this airplane located in airport E. The precision-recall curve for the validation images shows a much better performance, further supporting that this extremely bad performance only applies to the samples in airport E. An analysis of the imaging geometries for airplane #4 shows that those parked in airport E are imaged with a different orientation with respect to the SAR sensor, causing them to look completely different from those available in the training data.

Overall, even when adding more airplane classes and locations, the results for airplane #1 remain consistent with all the previous test cases: a near perfect detection performance is achieved for the "varied split", and a relatively small performance drop can be observed for the splits with different conditions for the test images (e.g., the "airport split" here). The results for airplane #2 for the "varied split" also show a near perfect performance, same as in the third test case. Finally, the results for airplanes #3 and #4 cannot be properly compared to those obtained in the third test case, as there were not enough test samples there.

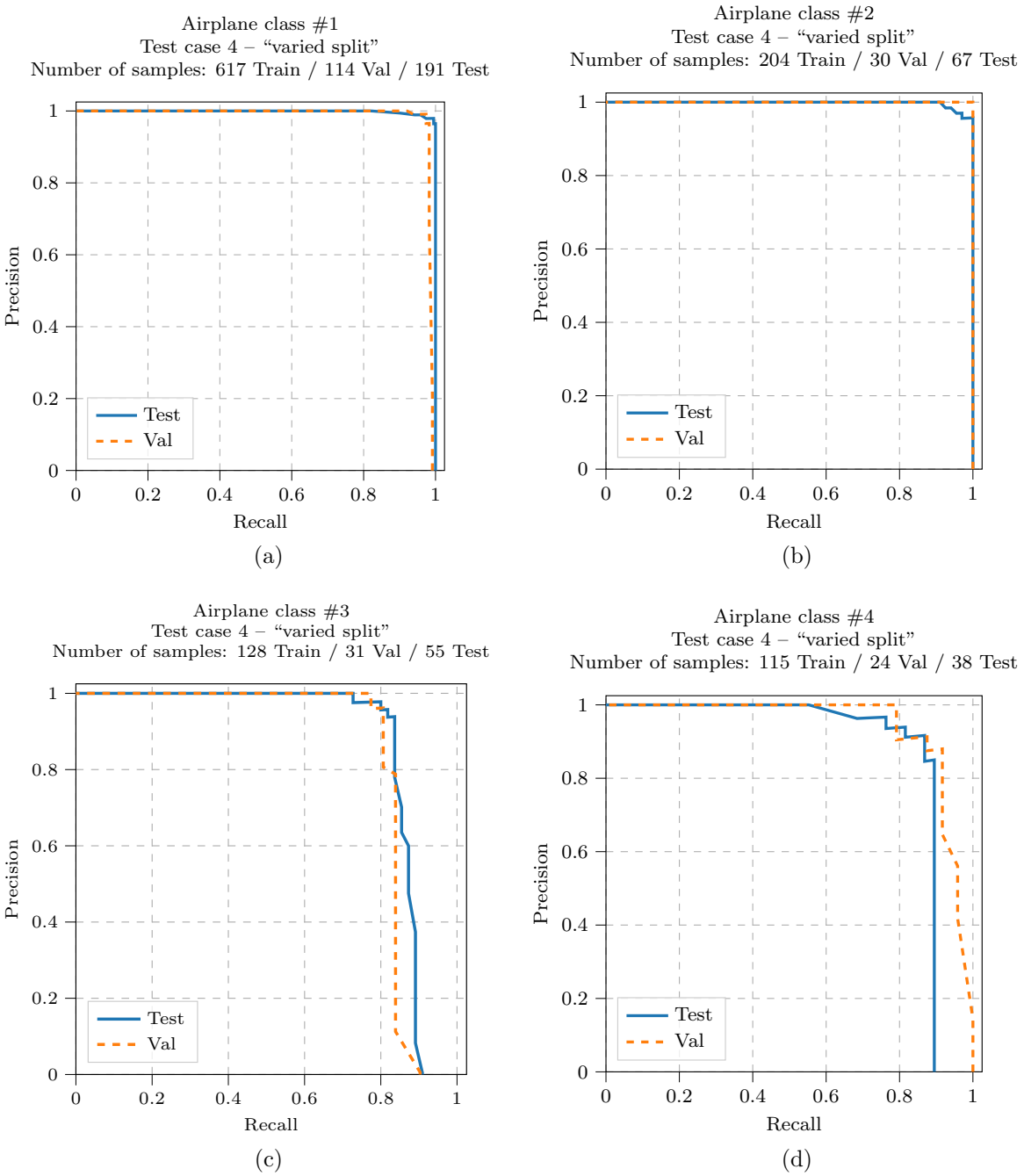


Figure 8.13: Precision-recall curves for the "varied split" of the fourth test case (all airplane types in all airports). Results for the different airplane types: a) #1, b) #2, c) #3, d) #4.

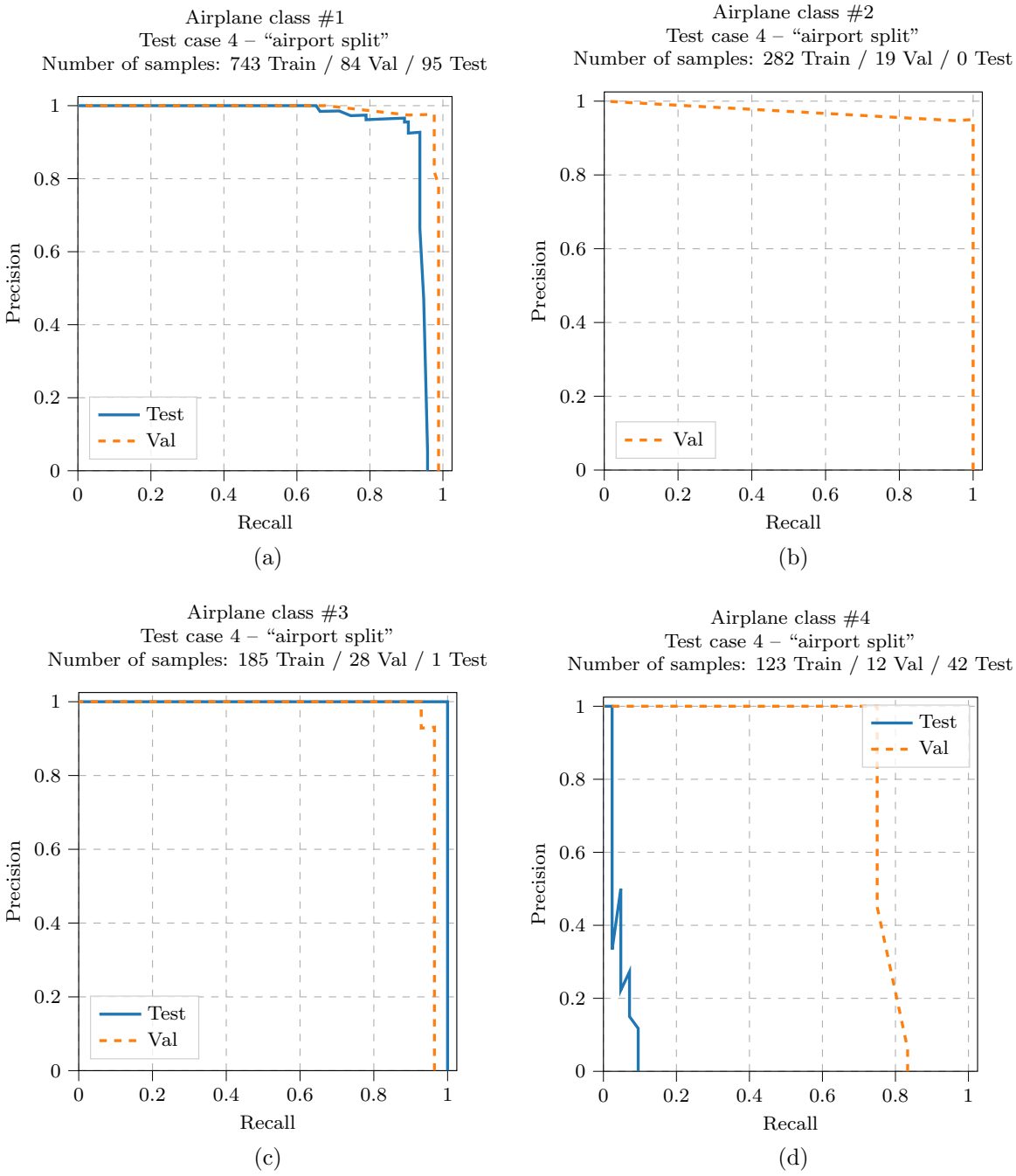


Figure 8.14: Precision-recall curves for the "airport split" of the fourth test case (all airplane types in all airports). Results for the different airplane types: a) #1, b) #2, c) #3, d) #4.

Table 8.5: Average precision values for the different test cases and airplane classes.

Test case	Split	AP for each class				Mean AP
		#1	#2	#3	#4	
One airplane type in a single airport	Varied	0.993	–	–	–	0.993
	Incidence	0.960	–	–	–	0.960
	Season	0.964	–	–	–	0.964
One airplane type in multiple airports	Varied	0.998	–	–	–	0.998
	Airport	0.967	–	–	–	0.967
All airplane types in a single airport	Varied	0.995	0.999	0.111	1.0	0.776
	Incidence	0.957	0.961	1.0	0.974	0.973
	Season	0.960	0.923	0.5	1.0	0.846
All airplane types in multiple airports	Varied	0.998	0.998	0.866	0.878	0.935
	Airport	0.936	–	1.0	0.043	0.660

Table 8.6: Number of training, validation and testing samples for each class in the different test cases.

Test case	Split	Number of samples for each class (train/val/test)			
		#1	#2	#3	#4
Single airport	Varied	275/40/93	204/30/67	13/1/9	25/3/7
	Incidence	255/40/74	186/28/60	19/0/2	23/3/6
	Season	196/26/51	135/19/37	14/1/2	18/2/4
Multiple airports	Varied	617/114/191	204/30/67	128/31/55	115/24/38
	Airport	743/84/95	282/19/0	185/28/1	123/12/42

8.3.2 Visual analysis of some example results

After this comprehensive analysis of the precision-recall curves and the corresponding AP scores, some of the obtained detection results will be shown here for a few example image patches. For this, the detected bounding boxes will be drawn in red color over the SAR images. The predicted airplane types and confidence scores will also be written next to the corresponding bounding boxes. Only the detections with a confidence above 33% will be shown. For easier verification, the annotated bounding boxes and the corresponding classes (i.e., the ground truth) will be represented in blue.

Some of the results obtained for the "season split" of the third test case can be seen in Fig. 8.15. The two image patches on the left column (Fig. 8.15a and 8.15c) were both taken from the same TerraSAR-X image, and show two different locations inside one airport. The two image patches on the right column (Fig. 8.15b and 8.15d) show these same two locations at a different time, as these were taken from a different TerraSAR-X image. The two TerraSAR-X images were acquired during winter time, with varying amounts of snow covering the airport. These four image patches illustrate the generalization capabilities of the proposed method, which correctly detected almost all the airplanes surrounded by snow, even though it was trained with images acquired in other seasons (and without snow on them). The predicted bounding boxes are very accurate, with their sizes and positions being nearly identical to those of the manual annotations, exhibiting a displacement of at most a few pixels. All the detections have also very high confidence scores, with the lowest one being 88%, but most of them being above 95%. The only errors in

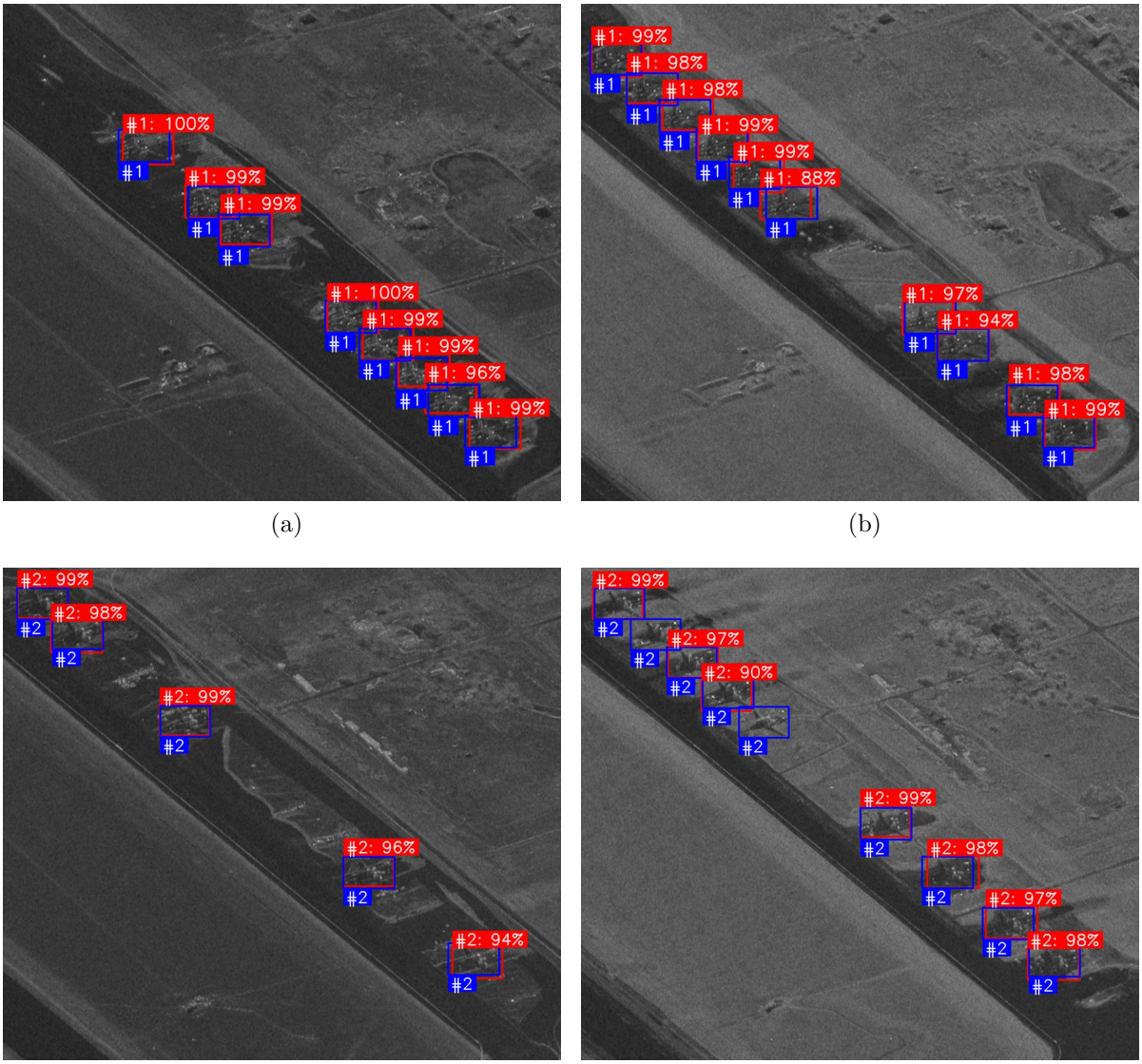


Figure 8.15: Example detection results for the test images of the "season split" of the third test case. The obtained bounding boxes, with the corresponding classes and detection confidence scores, are shown in red. For verification, the annotated bounding boxes and their classes (i.e., ground truth) are shown in blue. a-b) Results for two image patches showing the same location at two different times, with different amounts of snow. c-d) Results for another location, again shown at two different times.

this example correspond to two airplanes of type #2 which were not detected. These are located towards the top of Fig. 8.15d, which corresponds to the TerraSAR-X image with a higher amount of snow. These two airplanes are not only surrounded by snow but also partially covered by it, making their appearance even more different from those available in the training dataset.

While the examples in Fig. 8.15 illustrate the detection for a different season, these results only contain two of the four airplane types, and all the airplanes are parked in a row with the same orientation. More diverse examples can be seen in Fig. 8.16, which shows some of the obtained results for the "varied split" of the fourth test case. Here, the airplanes parked at different airports and with different orientations, and the other airplane classes can also be seen. Overall, the method also performs very well for these images, predicting accurate bounding boxes with a high confidence and the correct classes for almost all of the airplanes. Only two airplanes

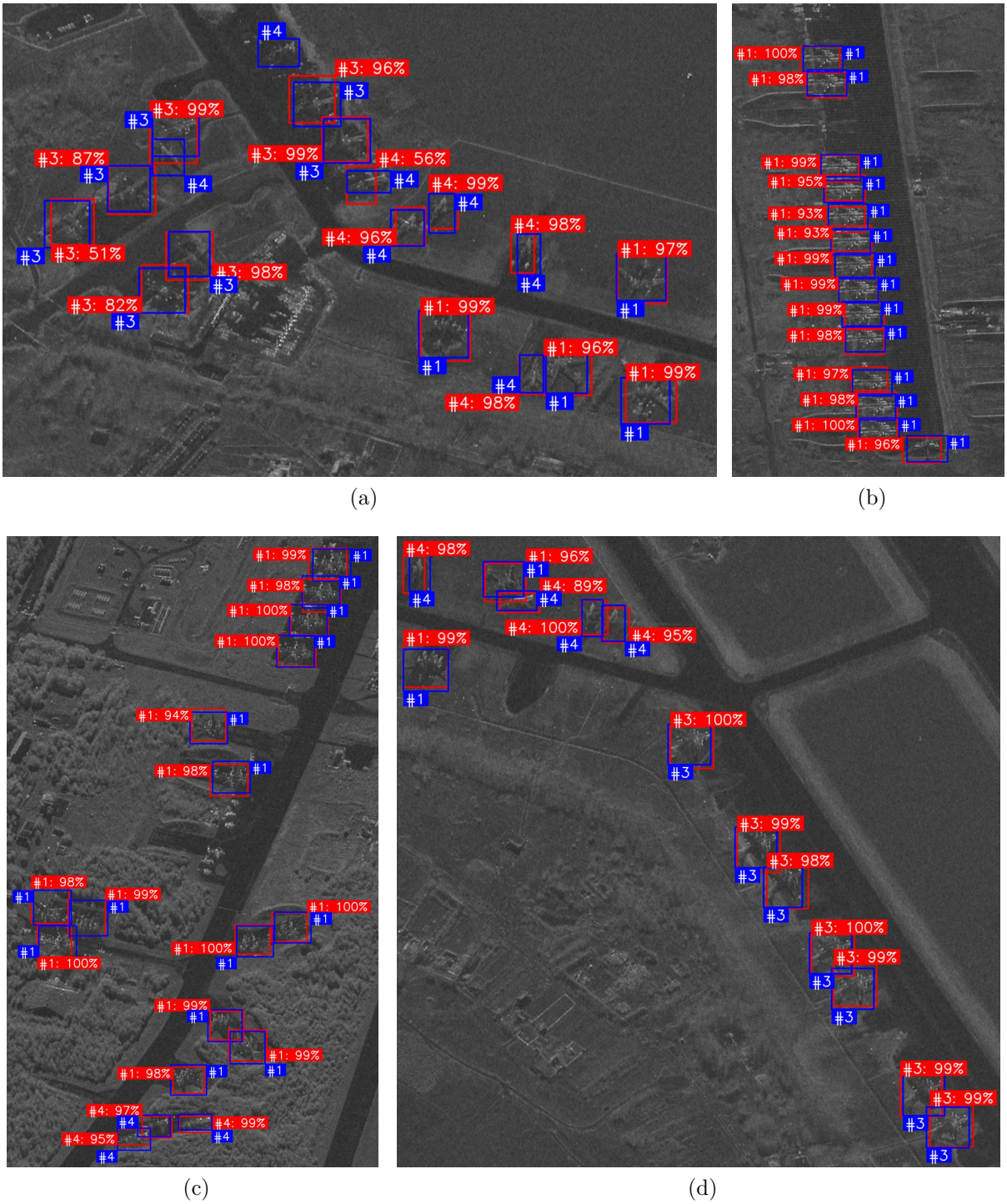


Figure 8.16: Example detection results for the test images of the "varied split" of the fourth test case. The obtained bounding boxes, with the corresponding classes and detection confidence scores, are shown in red. For verification, the annotated bounding boxes and their classes (i.e., ground truth) are shown in blue. The four image patches show different airports and airplanes types.

of type #4, located towards the top left part of Fig. 8.16a, could not be detected. Additionally, a couple of the correct detections, also in Fig. 8.16a, have relatively low confidence scores of 51% and 56%, and one of these has an inaccurate bounding box, indicating that this particular airplane was detected using a template acquired with a different imaging geometry.

8.3.3 Runtime analysis

Here, the runtimes for the performed experiments will be analyzed. For this analysis, two different phases will be distinguished: training and inference. The training phase, during which the network learns from the available training data, will be more time consuming, but will typically only need to be performed once. During the inference phase the trained network will be applied to detect objects in one or multiple SAR images. This operation is much faster, but it can potentially be repeated for a very large number of images. Below, the runtimes for both phases will be listed and analyzed separately. In all cases, the computations were performed using a single Nvidia RTX 3090 GPU.

Here, the network was trained multiple times for the different test cases and splits to analyze the generalization capabilities of the proposed method. Table 8.7 lists the time required for the different training runs. The training time depends mainly on the number of training epochs, because the same number of image pairs will be sampled during each epoch independently of the dataset size. While less relevant, the total training time will also be affected by the size of the validation set: to monitor the learning process, here the network was evaluated after each training epoch by performing inference in all the images of the validation set. For all the splits of the first three test cases, the network was trained during 10 epochs, and the training times ranged from 2 hours and 9 minutes (fastest) to 3 hours and 11 minutes (slowest). For the two splits of the fourth test case, the network was trained during 40 epochs (due to the higher complexity and the larger dataset), and the training times were significantly longer: 12 hours and 13 minutes, and 19 hours and 56 minutes.

The inference runtimes for the images in the validation and test sets of the different splits are listed in Table 8.8. These depend mainly on the number of images (also listed on Table 8.8), the image size, and the size of the template database. After the applied pre-processing the size of the TerraSAR-X images ranges from 107 to 239 million pixels, with an average size of 165 million pixels per image. The previously listed runtimes do not include the time required for pre-

Table 8.7: Training time for the different test cases.

Test case	Split	Training epochs	Total time
One airplane type in a single airport	Varied	10	2 hours 17 min
	Incidence	10	2 hours 9 min
	Season	10	2 hours 13 min
One airplane type in multiple airports	Varied	10	3 hours 11 min
	Airport	10	2 hours 44 min
All airplane types in a single airport	Varied	10	2 hours 30 min
	Incidence	10	2 hours 20 min
	Season	10	2 hours 10 min
All airplane types in multiple airports	Varied	40	12 hours 13 min
	Airport	40	19 hours 56 min

Table 8.8: Runtimes (given in seconds) for inference in the different test cases.

Test case	Split	Validation		Test		Average time per image
		Total time	Images	Total time	Images	
One airplane type in a single airport	Varied	35.29	3	57.81	7	9.31
	Incidence	31.63	3	49.67	6	9.03
	Season	29.12	2	38.99	4	11.35
One airplane type in multiple airports	Varied	121.23	7	218.91	13	17.00
	Airport	87.42	5	94.64	6	16.55
All airplane types in a single airport	Varied	53.31	3	102.36	7	15.57
	Incidence	39.82	3	73.84	6	12.63
	Season	32.56	2	45.66	4	13.03
All airplane types in multiple airports	Varied	233.91	7	418.98	13	32.65
	Airport	158.21	5	166.80	6	29.55

processing (e.g., despeckling), as this analysis was focused exclusively on the proposed method, and different speckle filters will have significantly different runtimes. To remove the influence of the different number of images in the validation and test sets of the different splits and test cases, the average time per image was computed using all the validation and test images for each split. The corresponding column of Table 8.8 illustrates how the inference time grows when the size of the template database increases (by adding more samples for different airplane types and/or imaging geometries).

8.3.4 Combination with change detection

As previously introduced, the proposed ATR method can be applied to determine how many airplanes, and of which types, are present in a given airport at different times. However, it cannot reliably provide information on the movement of airplanes (e.g., arrivals and departures), as it cannot determine whether an airplane which is parked in the same spot in two images moved or remained stationary. This information is certainly relevant: if the number of airplanes of each type does not significantly vary over time and these airplanes are typically parked in the same spots, the ATR results will provide little insight on the level of activity at the airport (e.g., most airplanes could remain stationary, or they could frequently move). This can be addressed by jointly applying the ATR and CD methods proposed in this thesis, and combining the complementary information provided by both methods.

The CD method presented in Chapter 5, which was previously demonstrated by detecting changes due to construction activity, can also be applied for this application. This method can detect changes due to the appearance or disappearance of man-made objects of any kind, and can also determine the time during which these remain coherent (and therefore static and unchanged). Below, in Fig. 8.17, an example of the application of the CD method for monitoring airplane arrivals and departures will be shown. For this example, the method was applied using the parameter settings listed in Section 7.2.2, except for the threshold for the CD metric, which was set to $\gamma_t = 0.35$. Here, a lower threshold was chosen because some of the performed experiments have shown that even changes in the airplane surroundings can cause their coherence to drop slightly (e.g., due to multiple reflections).

For this example, four SAR images acquired at different times and showing several parked airplanes were used. These can be seen in Fig. 8.17a through 8.17d. Here, T_i denotes the acquisition time of the i-th image in this time series. Visual interpretation of this image sequence clearly reveals that some of these airplanes moved during this time (e.g., the two airplanes at the top of Fig. 8.17b were present at T_2, and left sometime after T_2 and before T_3). Such obvious changes can also be directly identified by analyzing the ATR results and do not require applying a CD method. However, other changes cannot be so easily identified, as another airplane of the same type might be parked at almost the exact same position in the next image. This kind of changes can only be consistently detected by applying CCD.

All the changes in this image sequence can be better visualized in the multitemporal color composite images in Fig. 8.17e, 8.17e and 8.17g, which highlight the changes between the three consecutive image pairs. Each of these RGB composite images shows two amplitude images in the green and red channels, and the coherence of the corresponding image pair in the blue channel. Here, some changes which cannot be easily identified on the SAR image sequence become evident. For example, the airplane appearing in green, yellow and red colors towards the top of Fig. 8.17e indicates that one airplane was present in this location at T_1 and also at T_2, but that its position changed slightly during this time period. The cause for this movement cannot be distinguished: the same airplane might have remained on the ground and just moved slightly, or the airplane might have departed some time after T_1 and another airplane of the same type might have arrived before T_2. Here, it will be assumed that such changes are caused by a departure and a subsequent arrival, as this seems more likely.

The results obtained after applying the proposed CD method to these four images can be seen in Fig. 8.17h through 8.17k, where the segmented objects were overlaid over the SAR amplitude images, with the different colors representing the different time periods during which these objects remain coherent. Even though ground truth is not available, the accuracy of these results can be verified by visual interpretation, with the aid of the three previously shown color composite images. All the changes due to airplane movements seem to be correctly detected. Interestingly, a part of the airplane at the bottom remains coherent between T_1 and T_3, whereas another part only between T_2 and T_3, which appears to be a contradiction. This effect can also be seen in Fig. 8.17e and 8.17f. An analysis of these images suggests that the airplane did not move between T_1 and T_3, but that a change involving a nearby object caused the coherence to drop for some of its pixels, likely due to multiple reflections occurring between this object and the airplane. The CD results for this airplane are therefore not necessarily wrong, as it did indeed remain static between T_1 and T_3, but it was also indirectly affected by a change occurred during this time.

Because the CD method will detect changes associated to all kinds of man-made objects, its results should be combined with the detections of the proposed ATR method to identify only the changes due to airplane arrivals and departures. Figure 8.18 illustrates this process for this particular example. The ATR detections for these four images and the CD results inside the corresponding bounding boxes can be seen in Fig. 8.18a through 8.18d. A simple approach to combine both results and determine the time period during which the detected airplanes remained stationary would be to the find the most dominant color (i.e., the one covering a larger area) inside each of the bounding boxes. The results obtained by this approach can be seen in Fig. 8.18e through 8.18h, where the obtained temporal information was written next to the bounding box of each airplane. These results are correct for all the airplanes except for one: the airplane at the bottom of Fig. 8.18e, 8.18f and 8.18g. This exception corresponds to the previously described case where different parts of the same airplane remain coherent during different time periods. Because here the ATR and CD results were combined separately for each image, the results for this airplane are inconsistent across the time series. According to Fig. 8.18e, this airplane remained stationary

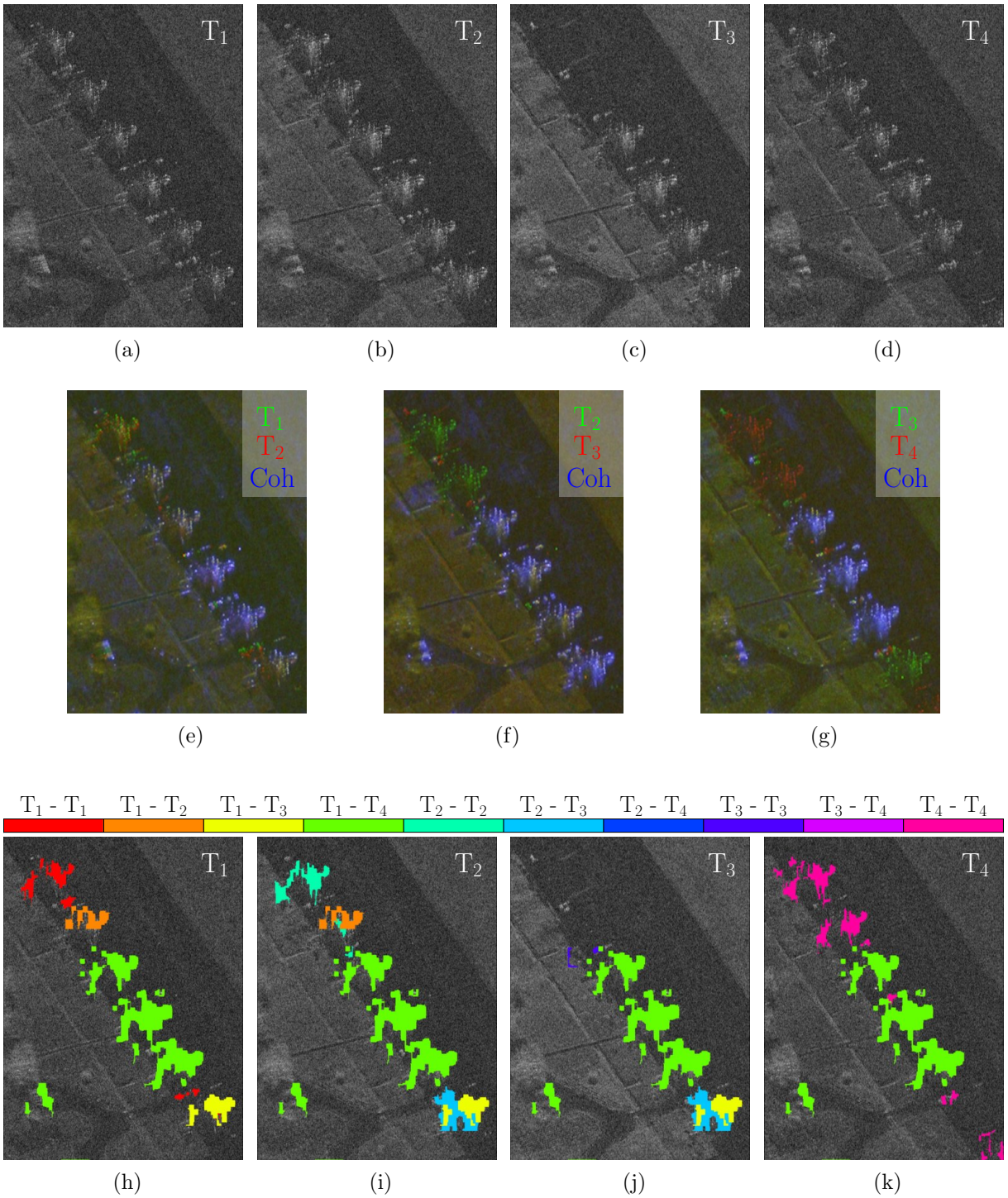


Figure 8.17: Example of the application of change detection for monitoring the arrival and departure of airplanes. a-d) Four SAR images acquired at different times, showing several parked airplanes. e-g) Multitemporal color composite images highlighting the changes between the three consecutive image pairs. h-k) Results of the proposed change detection method overlaid over the four original SAR images, with the different colors illustrating the images during which each object remained coherent (and therefore static and unchanged).

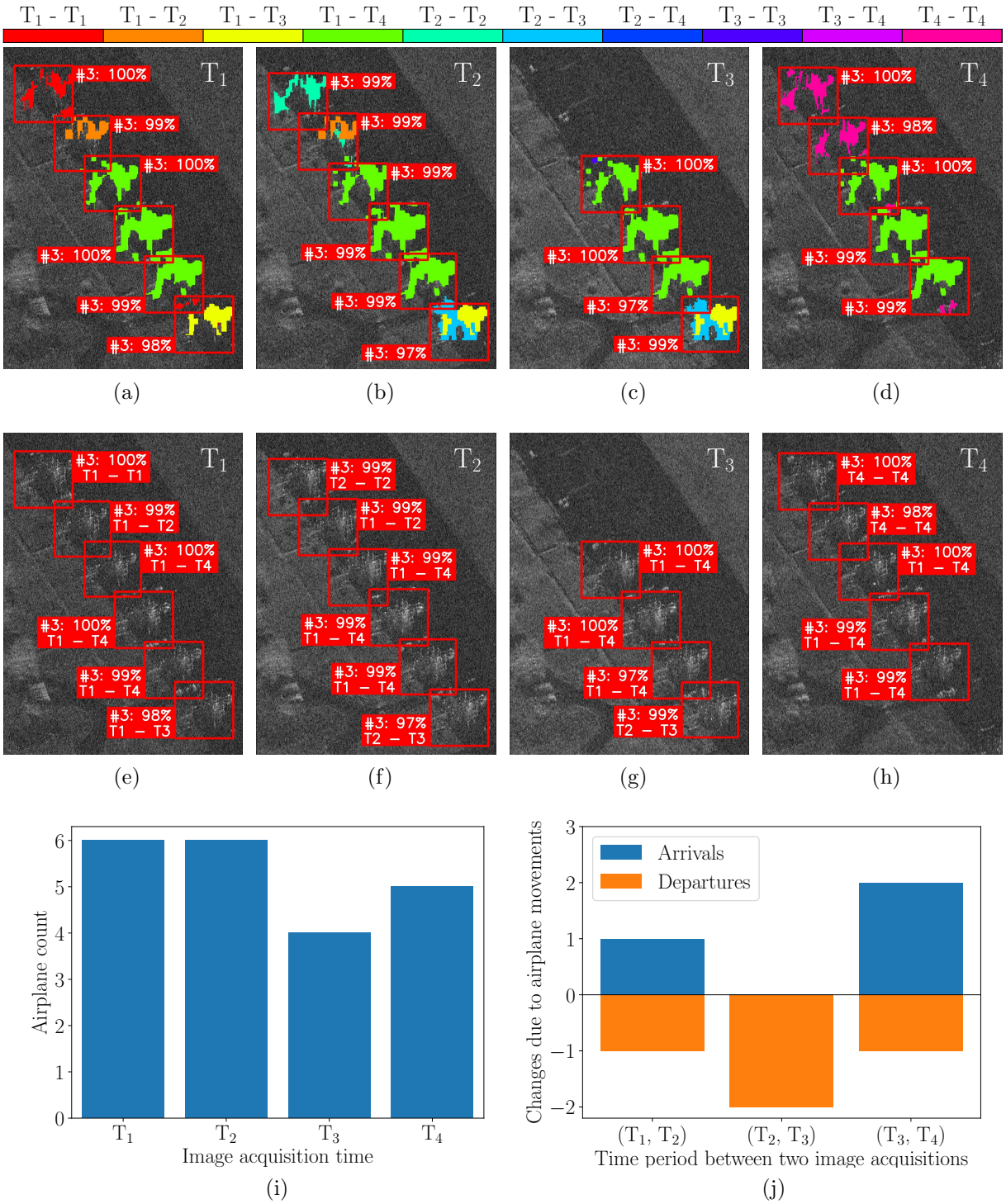


Figure 8.18: Example of the combination of change detection and automatic target recognition for the monitoring of airport activity, shown for the previous example of Fig. 8.17. a-d) Airplanes detected in each of the four SAR images, and change detection results inside the corresponding bounding boxes. e-h) Detection results with complementary information obtained from the change detection, indicating the time period during which each airplane remained stationary. i-j) Plots summarizing the resulting information: number of airplanes present at the acquisition time of each time (i), and changes in the number of airplanes caused by arrivals and departures between each of the three consecutive image pairs (j).

between T_1 and T_3, but according to Fig. 8.18f and 8.18g, it remained stationary only between T_2 and T_3 (implying that it moved between T_1 and T_2). Such contradictions could be addressed during a post-processing step, where all the information for the time series (i.e., the detections for all individual images and all the CD results) is combined in a consistent way. However, the development of more sophisticated methods for combining the ATR and CD results will not be addressed in this thesis, and remains a future research topic.

Finally, the information obtained by combining the ATR and CD results can be exploited to quantify the level of activity at an airport and how it changes over time. A statistical analysis could also be applied on this information to detect anomalies (e.g., periods of unexpectedly high or low activity). For this small example, the obtained information can be summarized in the two plots shown in Fig. 8.18i and 8.18j. The plot in Fig. 8.18i shows the number of airplanes at the acquisition time of each image, and the one in Fig. 8.18j shows the number of airplane arrivals and departures between each of the consecutive image pairs. In this small example, all airplanes are of the same type, but this information could also be represented and analyzed separately for the different airplane classes.

9 Discussion

The novel methods presented in this thesis were applied to real SAR data to solve specific problems involving the monitoring of different types of human activity, and the obtained results were shown in Chapter 8. In this Chapter, the outcomes will be discussed, and the suitability of the different methods for the intended applications will be analyzed. Part of the material in this Chapter has been published in [Villamil Lopez & Stilla, 2021] and [Villamil Lopez & Stilla, 2022].

9.1 Monitoring of oil storage

In Section 8.1, the proposed method for the automatic estimation of all the relevant parameters of oil storage tanks was applied to a time series with three TerraSAR-X images of the port of Rotterdam, containing 167 storage tanks of different sizes and types. The obtained results show that, as expected, the use of a time series increases the robustness of the method, reducing the number of tanks for which the method provides wrong results, especially for tanks with a fixed roof and/or smaller sizes. Additionally, for tanks with a floating roof, the use of a time series has another advantage: the relative changes in the amount of oil stored will be estimated much more accurately and robustly than when multiple images are processed individually. Because the estimation of the vertical displacements of the floating roof is independent from the detection of the semicircular double reflections, the relative changes will almost always be correctly estimated, even if there are errors when estimating the exact amount of oil or their maximum capacity.

The results have also shown that, if available, the use of some information regarding the approximate radius of each tank is of advantage. This reduces the size of the interval for the possible radius values, which leads to fewer errors when applying the custom Hough transform, and also makes the method significantly faster. In this work, these approximate radius values were obtained from OpenStreetMap, but these could also be obtained from the available imagery by applying some of the listed methods for the pre-detection step. Alternatively, rather than trying the many possible radius values covering a given interval, only a few specific values for the radius could be used, as most storage tanks have standard sizes.

When using a time series and the approximate radius information, the method performed very well, correctly fitting all the semicircular double reflections for 158 of the 167 storage tanks in the imaged scene, with errors in 3 of the 96 tanks with a floating roof, and 6 of the 71 tanks with a fixed roof. Overall, the method appears to perform better for tanks with a floating roof, which represent the most interesting use case, as the amount of oil stored in a given tank can only be determined if it has a floating roof. A possible explanation for the lower performance with tanks with fixed roofs is that their SAR signatures have fewer CSs which can be used for the detection of the semicircular double reflections. Additionally, for this particular dataset, the tanks with a fixed roof are mostly of smaller sizes, which further reduces the number of CSs detected in them.

A rigorous analysis of the accuracy of the estimates for the different dimensions could not be performed due to the lack of ground truth. Instead, the achievable accuracy was assessed for ten

storage tanks by comparing the obtained results with those derived from manual measurements. The estimates for the tank's height, radius and vertical position of the floating roof appear to agree between the different methods, but with small differences in the range of one to a couple of pixels. It is unclear whether one of these methods is more accurate than the other. On the other hand, the vertical displacements of the floating roof between consecutive images seem to be estimated much more accurately. This implies that the relative changes in the amount of oil stored in a tank with a floating roof will be estimated very accurately, whereas absolute measurements of the amount of oil stored in it will be slightly less accurate. This also applies to the estimated maximum capacity, both for tanks with fixed and floating roofs. The accuracy of these absolute measurements is also affected by the fact that, due to the side-looking imaging geometry of SAR sensors, an error on the estimated radius will induce an error on the vertical position of the floating roof.

A good performance in the classification of storage tanks according to their roof type (i.e., whether they have a fixed or a floating roof) has also been demonstrated. As expected, due to the use of just three simple but very informative features, the tested classifiers perform very well even with small training datasets, obtaining average F_1 scores above 0.9 with just 10 training samples. Here, the use of a time series also slightly improves the classification results, especially with a larger training dataset, achieving an average F_1 score of 0.97 with 100 training samples.

An analysis of the method's runtime has also been performed, showing that the method runs fast enough even on a standard laptop, making the processing of a large amount of SAR images containing lots of storage tanks feasible. When using a time series, the whole processing does not need to be repeated each time that a new image is acquired. The new image can be simply co-registered and added to the existing image stack, updating the change detection results. Because the location, radius and height of the storage tanks will very rarely change, only the displacements of the floating roof and its initial position need to be recomputed.

Overall, the proposed method seems to perform very well, especially when a SAR time series is available and the used pre-detection method provides some rough information on the radii of the detected tanks. The obtained results suggest that this method could be used in practice for the intended application: the automatic monitoring of oil inventories using SAR time series of different refineries.

9.2 Monitoring of construction activity

In Section 8.2, the proposed method for the detection of changes associated with man-made objects was applied to a long time series with 49 TerraSAR-X images of Munich for the monitoring of construction activity. The performed experiments show that the method performs well and can accurately detect the changes associated to man-made objects. Seasonal changes (e.g., snow, changes in water level, vegetation, etc.) are ignored and do not result in false alarms. The capability to identify and ignore transient changes, where an external factor temporarily affects a man-made object which does not actually change, has also been shown. Most of the buildings in the imaged scene were affected by transient changes at several points during this long time series, as their roofs were covered with snow in some of the winter acquisitions, which caused their coherence to drop significantly. Nevertheless, the proposed CD metric, which exploits the coherence matrix, could ignore these irrelevant transient changes, correctly identifying which buildings remained unchanged.

Overall, the results indicate that the proposed CD method is well suited for the monitoring of construction activity. The proposed temporal analysis of the detected changes can successfully identify the changes due to newly constructed and renovated buildings, and determine the time

when the construction work finished. As expected, a relatively accurate segmentation of the changed objects can be achieved by exploiting both temporal and spatial information. Even a complex renovation process like that of the museum "Alte Pinakothek", where different parts of the same building were gradually renovated over a long time period, could be accurately monitored. The results for this and other buildings were in agreement with the changes that could be observed in sequences of optical and/or SAR images, demonstrating the method's accuracy.

It is important to note that to identify changes to buildings using this method, a minimum of three images are required: one acquired before or during the construction work, and two acquired after the construction work was finished and at least certain time apart. Because of this, the changes due to the construction of two new buildings in the TUM area could not be identified, as the construction work was not completely finished by the time the last image in the dataset was acquired. Also, because the proposed temporal analysis only identifies the final changes to buildings, the changes due to newly constructed buildings or renovations to existing ones cannot be distinguished. Some of the obtained results indicate that these two types of changes have a different characteristic behavior, and could potentially be distinguished by analyzing the evolution of the number of CSs inside the corresponding area. However, such analysis would also require a few images acquired before and during the construction work.

When applying the method for detecting changes due to the renovation of existing buildings, the combined use of multiple time series acquired with different imaging geometries (e.g., different orbit or look direction) should be considered. Otherwise some changes cannot be detected, as all the building façades cannot be imaged with a single imaging geometry.

For the build-up of festivals and similar events involving many small objects close together and constantly changing, the proposed method is not able to separate individual objects using neither spatial nor temporal information. In such cases, closely packed objects are grouped together when performing the object-based analysis. Other than that, the change segmentation works fairly well.

Even though the proposed CD method was demonstrated by detecting changes due to construction activity, it can also be used for other applications. This method can detect changes due to the appearance or disappearance of man-made objects of any kind, and can also determine the time during which these remain coherent (and therefore static and unchanged).

The CD method presented in this thesis was evaluated in a rather qualitative way, as quantitatively evaluating its performance and comparing it to other CD methods would require a dataset with ground truth. A suitable and publicly available dataset could not be found, and generating one is not an easy task. Acquiring ground truth or manually labelling the data for such a CD task is very challenging and time consuming. Besides, time series of spaceborne SAR data with such a high resolution are typically not freely available. Also, the generation of realistic synthetic data does not seem feasible, as the proposed method uses SLC SAR images and exploits their phase.

The proposed CD method was developed for a specific task: detecting changes associated with man-made objects. For this task, this method is expected to perform better than general-purpose CD methods. However, this specificity limits its applicability: it is not well suited for applications where changes to natural targets are relevant. Also, it will perform badly when applied to data with a much lower resolution (e.g., Sentinel-1), as the CS detection requires a large bandwidth to work properly. On the other hand, better performance can be expected when using data with even higher resolution. The proposed method can work with long temporal baselines, but the potential of the applied temporal analysis increases with the temporal resolution of the used time series. This makes it especially interesting for SAR missions involving large constellations with

very high revisit, like those currently being built by some of the NewSpace companies which were listed in Table 1.1.

9.3 Monitoring of airport activity

In Section 8.3, the method for object recognition proposed in this thesis was applied to detect and classify airplanes of four different classes, and its performance was evaluated using a dataset with 60 TerraSAR-X images of five different airports. The performed experiments have shown that the proposed modifications to the selected SiamFC++ network architecture, which was originally designed for a different task, result in an improved performance. Among these, the strategy for sampling the training data was especially relevant. The application of a despeckling method during pre-processing was also beneficial, and an even better performance can be expected when using a state of the art despeckling method. An analysis of the obtained results has shown that under most circumstances, this method performs very well for the intended application, with just a few exceptions which are mostly related to the available data.

For the ideal application scenario where the method is applied to images acquired with similar conditions to those used for training, denoted here as "varied split", the results suggest that an extremely good detection performance can be expected. The only exception corresponds to the airplane type #3 in the third test case, for which very few test samples were available, making the computed AP score unreliable. If the object classes with very few test samples (e.g., less than 10) are ignored, the mean AP scores for the "varied split" of the four test cases become 0.993, 0.998, 0.997 and 0.935. These values indicate a near perfect detection performance for the first three test cases, and a slightly lower, but still very high performance for the fourth and most difficult test case.

An analysis of the results of the other splits indicates that the performance will drop slightly when the test images are acquired under slightly different conditions, but the method can still perform very well in these situations. This can be seen by the fact that, after excluding the unreliable AP scores for the classes with very few test samples, a mean AP well above 0.9 is obtained for all cases except from one. This exception corresponds to the "airport split" of the last test case, where a very good detection performance was achieved for one of the classes (#1), and an extremely bad performance for the other (#4). An analysis of the imaging geometries available for this particular case revealed that the airplanes of class #4 in the test airport were parked with orientations completely different from those available in the training data. These results suggest that the method can generalize well up to a certain point, with the performance dropping minimally for small differences in the data. However, for very large differences, especially in the object orientation, it seems that the detection performance will be significantly affected.

The visualization of some of the detection results showed that, for most airplanes, the detections have very high confidence scores and the predicted bounding boxes are very accurate. Most bounding boxes exhibit a displacement of at most a few pixels with respect to the manual annotations, and mostly identical sizes. Most of the errors correspond to false negatives (i.e., airplanes which were not detected), with false positives being less common. Misclassification errors, where an object is detected but assigned a different class, are extremely rare. The detection results also include an estimation of the airplane orientations. This information was not visualized nor analyzed here, as it is not that relevant. Nevertheless, the estimated orientations appear to agree with the annotated ones for most airplanes, with errors in the order of a few degrees.

A comparison of the results obtained for the different test cases suggest that training the network for a more specific task (e.g., for a single airport and/or airplane type) provides no significant advantage. The results for most of the performed experiments indicate that increasing

the size of the dataset by adding additional object classes and/or locations does not significantly affect the performance for the previously available data. Nevertheless, it is important to note that the object classes and locations used for these experiments are quite similar (with all the object classes corresponding to different airplanes, and the different locations to airports). This trend might not hold when including much more diverse data with completely different types of objects and scenes (e.g., ships at ports).

The performed runtime analysis shows that the network can be trained in a reasonable amount of time (e.g., a few hours) using a single desktop GPU. The results of some of the experiments performed in Chapter 7.3.2 indicate that the training time could be significantly reduced without adding more hardware by training for less epochs, with only a small impact on performance. The results also demonstrate that the inference time is also reasonable, ranging from 10 to 30 seconds per image (for TerraSAR-X images with approximately 150 million pixels). The inference time increases almost linearly with the size of the template database, which will increase with the addition of more object classes or imaging geometries. This represents a disadvantage of the proposed method with respect to traditional object detection methods, for which the runtimes will not significantly increase with the number of object classes. For the proposed method, this could potentially become a problem if too many (e.g., 100) object classes are added. However, this could be partially addressed by using prior knowledge about the imaged scenes (e.g., from OSM), to limit the number of object classes to be detected in a given location. In addition to improving the inference runtime, this could also help to eliminate some of the false detections.

The potential of combining the results from the proposed ATR and CD methods was also illustrated with an example, demonstrating that the two methods complement each other. The ATR method can be applied to individual images or series of images acquired with different imaging geometries, and can accurately detect and classify airplanes. However, being a supervised learning method, it requires a certain amount of training data. On the other hand, the CD method can detect changes due to the movement of man-made objects inside the airport without any training data, but it requires a time series with repeat-pass images and cannot distinguish which changes correspond to airplanes. By jointly applying both methods, the arrival and departure times of the detected airplanes can be estimated, and changes caused by the movement of other unknown objects (e.g., maintenance trucks) can also be detected. By itself, the ATR method can determine the number of airplanes present at the acquisition time of each image, but when used in combination with the CD method, the number of airplane arrivals and departures between each of the consecutive image pairs can also be determined. This additional information can provide valuable insights on the level of activity at an airport. The simple strategy used in this thesis to combine the ATR and CD results seemed to work well in most cases, with the exception of one airplane, for which the results were inconsistent across the time series. A more sophisticated post-processing should be applied to properly combine the ATR and CD results for the complete time series in a consistent way, and even correct some errors on the ATR or CD results. In conclusion, a combination of these two methods seems very promising, and could also be applied to monitor other types of objects.

10 Conclusions and outlook

This Chapter concludes this thesis, presenting the most important conclusions by answering the research questions outlined in Section 1.5 based on the results of the performed experiments, and proposing future research directions to continue this work.

10.1 Conclusions: answering the research questions

How robustly and accurately can a method using high resolution SAR images automatically estimate the amount of oil in storage tanks?

The experiments performed on a dataset with 167 oil storage tanks of different types and sizes indicate that, as long as a proper method is used to process the VHR SAR data, a very robust estimation is possible. When applying the best variant of the method proposed in this thesis to estimate the amount of oil stored in the tanks with a floating roof, which represents the most interesting use case, correct estimates were obtained for 93 out of the 96 tanks in the dataset. The results for the remaining three floating roof tanks contained only small errors, leading to a bias in the estimates of their initial fill levels. The radius and the vertical displacements of the floating roof, and therefore the changes in the amount of stored oil, were still correctly estimated for all three. Nevertheless, even these small and rare errors could potentially be avoided by using a pre-detection method providing more accurate information on the approximate location and size of the oil tanks, which should be feasible, as a relatively high uncertainty was assumed in the performed experiments. Knowledge of the most common tank sizes (i.e., radius and height) could also be potentially taken into account to make the method more robust. The method was only tested using SAR images from a single refinery, but it should perform similarly in other refineries as long as they have similar storage tanks, which is to be expected, as these tanks are built following certain standards. Also, external factors such as different sunlight illumination or atmospheric conditions (e.g., clouds, fog or smog), which can introduce additional errors for methods using optical images, will have no effect on the estimations obtained from SAR images.

A rigorous analysis of the achievable accuracy would require ground truth data for the precise dimensions of the imaged oil tanks, as well as the amount of oil stored in them at the times the images were acquired, but such data was not available and cannot be easily obtained. Nevertheless, the performed experiments and the obtained results suggest that very accurate measurements can be performed using VHR SAR images. The measurement principle, which involves deriving a tank's dimensions from its semicircular double reflections, has been validated in the literature; and the method proposed in this thesis can accurately detect these double reflections. The achievable accuracy will depend mainly on the spatial resolution and the chosen imaging geometry. The resolution of the TerraSAR-X data used in this work already enables highly accurate measurements, and newer spaceborne SAR sensors can acquire images with an even higher resolution. Because SAR sensors can acquire images under all conditions, a well suited imaging geometry can be freely selected when planning the image acquisitions. Using a steep incidence angle will allow to derive the heights from the layover with a higher accuracy, and also to see better inside the

oil tanks. The accuracy of the performed measurements could be further improved by exploiting the fact that storage tanks often have standard sizes, rounding the results for the tank radius and height to the closest standard values. This could help to get a precise estimation of a tank's radius, which is of importance because, due to the measurement principle, an error in the estimated radius will induce a larger error in the estimated vertical position of the floating roof. All things considered, it seems that very precise relative measurements (involving the changes in the amount of oil stored) can most certainly be achieved, whereas absolute measurements of the exact amount might contain a certain bias. This bias is related to the possible presence of a foundation of unknown height below the tanks, and an error in the estimation of the initial position of the floating roof (possibly due to an error in the radius estimate). A more thorough analysis of the achievable accuracy, as well as the development of strategies to correct this possible bias, are two topics that will be addressed in future research.

To what extent can an unsupervised method exploiting SAR time series identify the changes corresponding to specific types of objects or events?

Unsupervised change detection (CD) methods using VHR SAR data are very useful for the monitoring of human activity, as they can extract valuable information from SAR time series and, unlike automatic target recognition (ATR) methods, do not require any training data. Most CD methods simply result in a binary change map highlighting all the changes, but many practical applications require the detection of the changes corresponding to specific types of object and/or events. As shown in this thesis, an unsupervised CD method can identify certain types of changes by taking into account their scattering mechanism, their temporal behavior and, if available, prior knowledge about the imaged scene.

The CD method presented in this thesis exploits the different scattering mechanisms of man-made objects (typically contain many strong point scatterers) and natural targets (containing mostly distributed scatterers) to detect only the changes involving man-made objects while ignoring those to natural targets (e.g., vegetation). Changes involving the appearance or disappearance of many closely packed CSs are very likely to correspond to man-made objects. As a side benefit, this also enables coherent change detection even with long temporal baselines, as CSs are not significantly affected by temporal decorrelation.

After the changes involving CSs are detected, the applied spatio-temporal analysis allows to identify changes of certain sizes and/or with a characteristic temporal behavior. For example, this temporal analysis has been successfully applied in this thesis to identify changes due to the construction of new buildings or renovations to existing ones, as these imply the appearance of new CSs which then typically remain unchanged for a very long time. Additionally, this temporal information also helps to segment the changed objects, making it easier to separate an them from other objects in their surroundings. As long as two objects remain static and unchanged during a different time period, the temporal information alone will suffice to separate them, without the need of a sophisticated segmentation algorithm. On the other hand, if two objects are located next to each other and remain unchanged during the same time period, the proposed method will most likely not be able to separate them, and will consider them as a single object.

Finally, prior-knowledge about the imaged scene, which can often be obtained from other types of data (e.g., freely available map data like OSM, optical imagery, etc.), can also be exploited. Such information could be used to delimit the possible locations where some objects can be present or where certain events are expected to take place. For example, ships will be located on water, airplanes inside airports (e.g., on aprons, taxiways, runways, etc.), cars on roads and parking areas, etc.

In conclusion, changes corresponding to specific types of man-made objects can be identified if they exhibit a temporal behavior different from that of other objects in their surroundings, and/or if these objects appear only in specific locations. Additionally, changes that are either too small or too large can also ignored. However, this will not always suffice, and sometimes it will not be possible to reliably identify changes associated to certain objects among all the detected changes. For example, the proposed method can accurately detect changes due to the movement of man-made objects of a certain size inside an airport, but cannot determine which of these changes actually correspond to airplanes, as other man-made objects might also move and/or change inside the same area. Nevertheless, even if very specific changes (e.g., arrival or departure of airplanes) cannot always be unambiguously identified, this kind of analysis will allow to discard most of the irrelevant changes (e.g., seasonal vegetation changes, snow, other objects outside of the airport, etc.).

How much training data is required to accurately and robustly identify specific objects in VHR SAR images using a deep convolutional neural network, and how well can the trained network generalize to different locations or imaging geometries?

While a general statement on the required amount of training data cannot be derived, the experiments performed in this thesis provide some valuable insights on this. The results for the smallest split of the first test case, containing just 196 training samples of a single airplane class across 14 TerraSAR-X images, show that it is possible to successfully train the proposed network architecture with a relatively small amount of data, achieving a good detection performance. The results for the different splits of the third test case also show that the network could be successfully trained for the detection of four different airplane classes, with a total number of training samples ranging from 400 to 500 across the same 14 TerraSAR-X images, and with less than 20 samples for the less common class. Using the proposed network architecture and training strategy, a relatively deep CNN can be trained from scratch (i.e., without any pre-training) even with a small dataset. However, pre-training the CNN used for feature extraction on a large dataset like ImageNet has also been shown to significantly improve the performance, especially for larger networks. Interestingly, despite the reduced number of training samples, the network never seemed to overfit, even when training during a long time and using different learning rates. In all cases, the performance for the images in the validation set stopped improving at some point, but did not begin to degrade. This is most likely due to the fact that the network takes an image pair as an input. Because of the way that these pairs are generated, even for small training datasets, the network will rarely see the same inputs repeatedly during training.

The main focus of the experiments carried out in this thesis, using multiple test cases and different splits for each test case, was to perform a comprehensive analysis of the generalization capability, to understand how much the detection performance will be affected when certain conditions are different in the training and test data. The obtained results show that when the method is applied to data exhibiting different seasonal effects (e.g., snow) or acquired with a different incidence angle, the performance will drop slightly, but the method can still perform very well in these situations. As expected, the bigger the difference, the worse the detection performance will be. This can be seen by the fact that, for the "season split", most of the errors occur for the images with more snow. It is also important to note that the impact of different seasonal effects and incidence angles were analyzed in isolation, and a combination of both will most likely have a bigger impact on the detection performance. The results for the "airport split" suggest that applying the method to images from a different location can (but not necessarily will) affect performance in a very significant way. Its impact will mostly depend on some of the differences that the new location implies (e.g., objects being imaged from a different orientation, located on different surfaces, etc.), and how much these alter the appearance of the imaged

objects. Overall, the orientation of the objects with respect to the radar's line of sight has clearly the biggest impact on performance: a change in an object's orientation can completely change its SAR signature, making it very difficult to detect objects for orientations which are significantly different from those available in the training set. Unlike the incidence angle used to acquire an image or the season when the image is acquired, the orientation of the objects in the imaged scene will often be unknown, making it more difficult to account for this. However, the effect that imaging a given scene with a different orbit and/or look direction has on the orientation of the objects in it can and should be considered.

In conclusion, a network such as the one presented in this thesis can be trained to perform object detection and classification using a relatively low amount of training data, and it can also generalize well up to a certain point. This is of great importance because large and high quality datasets for object detection and classification in VHR SAR images are scarce, and such VHR spaceborne SAR data is also rarely openly available. Besides, the creation of such datasets by manually annotating objects in VHR SAR images is difficult (possibly requiring human operators experienced in the analysis of SAR images) and time consuming, which also makes it expensive. Nevertheless, the training data still plays a very important role and heavily influences the method's performance. However, the results of the performed experiments suggest that, rather than the dataset size, the quality and diversity of the training data are more important. The used dataset has lots of nearly identical and redundant samples for some of the airplane classes (e.g., showing the same object, imaged with the same imaging geometry and under similar conditions), which seem to add little value, as the performance does not improve further when adding more of these samples. On the other hand, the few exceptional cases were the method performed badly were mostly due to the lack of training samples for certain object orientations. Therefore, when creating a dataset, the focus should be placed on acquiring a few samples for the different imaging geometries (and especially the object orientations) for each class.

How can the temporal information obtained from change detection complement and improve the performance of other methods which typically analyze a single image?

The ability to acquire long time series of a given location, with images acquired at regular intervals and with the same imaging geometry, is one of the most powerful capabilities of EO satellites. For SAR satellites this capability is especially interesting, as SAR time series can be coherently processed to extract valuable information which, among other things, enables the detection of very subtle changes. Even when applying methods for the analysis of a single SAR image, some prior images will often be available, and the information in this time series can potentially be exploited in a beneficial way. In this thesis, this has been illustrated with two different examples, by using the temporal information provided by the proposed CD method to complement and improve the performance of the two other presented methods.

For the monitoring of oil storage, a simplified CD method was applied to identify the static and moving parts of the oil storage tanks. This temporal information is not really required by the proposed method, which can estimate all the relevant parameters of a storage tank from a single SAR image. For monitoring changes in the amount of stored oil, multiple images acquired at different times (and potentially with different imaging geometries and/or sensors) could also be processed separately, with each image providing a separate measurement. However, as demonstrated in this thesis, when the temporal information is included and properly taken into account, the method becomes more robust, with errors for an even lower number of tanks, and the accuracy of the measurements also increases. Interestingly, the biggest improvement in accuracy is achieved for the measurements involving the floating roof, which moves in between image acquisitions, and not for those involving the outer tank structure, which remains static.

When applying the proposed ATR method for the monitoring of airport activity, the joint analysis of the ATR and CD results provided valuable complementary information on the arrival and departure times of the detected airplanes, which could not have been obtained otherwise. The combination of all the evidence provided by the CD and ATR methods for the whole time series could also potentially allow to correct some wrong detections (e.g., one airplane only detected in a single image, but with the coherence indicating that it remained stationary). This was not demonstrated in this thesis, and more sophisticated methods for the combination of the ATR and CD results still need be investigated in the future. Furthermore, the capability of the CD method to detect changes due to the movement of unknown objects, including those not detected by the ATR method, could also be exploited to progressively increase the size of the dataset, as it will identify potential objects of interest (possibly corresponding to new classes, or missing imaging geometries for the existing classes) which could then be annotated by the users. A similar principle could also be applied when building a new dataset from scratch (before any deep learning method can be trained or used to perform ATR): once the objects in an image are manually annotated, the CD method could be applied to assist the annotation of future repeat-pass images, making the process faster. For example, unchanged objects could be automatically annotated in the subsequent images, and new objects could be marked and shown to humans for annotation.

10.2 Outlook

10.2.1 Increased potential with new SAR missions

As previously introduced, the current situation, with a rapidly increasing number of SAR satellites in orbit, is often referred to as the "Golden Age of SAR". The cost of manufacturing SAR satellites with VHR capabilities is sinking, and private companies plan to launch large constellations with dozens of satellites. These trends are expected to make it both easier and cheaper to access VHR SAR data, which will in turn increase the demand for methods like the ones presented in this thesis. These new missions will also enable the acquisition of images with an even higher spatial resolution, as well as time series with a much higher temporal resolution. These improved capabilities, and their implications on the performance of the proposed methods, will be briefly outlined below.

Newer SAR sensors are expected to deliver images with better range and azimuth resolution. In contrast to the 300 MHz bandwidth of TerraSAR-X, the new satellite design recently presented by Capella will have a bandwidth of 700 MHz*, and Umbra satellites are expected to have a bandwidth of 1200 MHz†. Therefore, in the near future, the range resolution will most likely improve by a factor of 2 to 4 with respect to the data used in this thesis. Also, new SAR satellites tend to be smaller and highly maneuverable, allowing for longer integration times: satellites from Capella and ICEYE can perform acquisitions in Spotlight mode with integration times of around 25 seconds [Castelletti et al., 2021; Muff et al., 2022]. This will result in an even better azimuth resolution than the 25 cm of the TerraSAR-X Staring Spotlight data used in this work. The improved spatial resolution will enable the detection and monitoring of smaller objects, and the performance of all the methods presented in this thesis will also improve. The CS detection, which plays an important role on the methods for monitoring oil storage and for change detection, works better at higher resolutions, and a higher density of detected CSs can be expected. Also, the spatial analysis performed by the CD method, involving the clustering of CSs and subsequent change segmentation, will benefit from the higher spatial resolution. Finally, more accurate

*https://www.capellaspace.com/capella-space-unveils-next-generation-satellite

†https://umbra.space/blog/umbra-hits-regulatory-jackpot-for-its-satellite-constellation Last accessed on February 5th, 2023.

measurements of the dimensions of the oil storage tanks will become possible, and the detection performance of the proposed ATR method will become even better.

New large SAR constellations will have a repeat-pass interval in the order of several hours, enabling multiple coherent acquisitions per day. Currently, the ICEYE constellation can already acquire repeat-pass time series with a temporal resolution of one day [Muff et al., 2022]. This represents an improvement by more than an order of magnitude with respect to the temporal resolution of 11 days of the TerraSAR-X time series used in this thesis. This will have a significant impact on the kind of information that can be extracted from these time series, making it possible to detect more and faster changes, and enabling additional applications. The CD method proposed in this thesis will also greatly benefit from such a high temporal resolution, as the performed temporal analysis will become more powerful. The time interval during which each object is present in the imaged scene and remains unchanged will be determined much more precisely, which will in turn make it easier to separate and segment objects by taking into account their different arrival and departure times.

10.2.2 Future research work

The three methods presented in this thesis show a very promising performance, indicating that they could be used in practical applications. Nevertheless, certain aspects can still be improved. Below, some of the topics which could be further investigated are outlined.

For the monitoring of oil storage, a more rigorous verification of the accuracy achievable by the proposed method should be performed. For this, ground truth data should be obtained for several storage tanks (including their dimensions and the amount of oil stored in them at different times), and SAR images of the same scene should be acquired at these same times. If during this validation a bias were to be found in some of the estimates, the ground truth data could also potentially allow to derive some correction factors to try to compensate this bias. Additionally, the potential of combining SAR images acquired with different imaging geometries (e.g., different orbits and/or incidence angles) could also be investigated. While the proposed method can already be applied separately to multiple images or time series acquired with different geometries to achieve more frequent observations, two such images could also be exploited jointly to improve the accuracy of the estimates for each tank's radius and height. Such improved estimates could be achieved by fitting the two semicircular double reflections of the outer tank structure jointly using both images, adding in this way an additional geometric constraint. The ability to perform more accurate measurements for the radius will be especially beneficial because, as previously explained, an error in the radius will induce a larger error in the estimated vertical position of the floating roof.

For the unsupervised change detection method, the possibility of exploiting multiple images jointly for CS detection could be explored, instead of performing the detection separately in each image and then applying a consistency check as a post-processing step. The segmentation of objects present in a single image of the series could also potentially be improved by using a more modern amplitude CD metric and/or a more advanced segmentation method. Additionally, a more sophisticated analysis of the segmented objects could be implemented to better distinguish different kinds of changes. For example, changes due to new buildings could potentially be distinguished from those due to renovations by analyzing the evolution over time of the number of CSs, as it was briefly shown in this thesis. Finally, this method, which has been applied here for monitoring construction activity and the detection of airplane arrivals and departures, could also be evaluated using datasets of different scenes to detect other kinds of changes.

Regarding the object recognition method, its capability to perform few-shot or even one-shot learning should be investigated and evaluated. The proposed method should be well suited for this, as the selected network architecture was originally developed for tracking arbitrary objects in videos given a single sample. Ideally, it should be possible to apply the proposed method in a similar way, enabling the detection of arbitrary objects (potentially of classes not seen during training) in SAR images by using new templates, without any additional training. However, the dataset used here, with only a few hundred samples of four object classes, is much smaller and less diverse than those used for video tracking, and might not suffice to learn how to compare arbitrary objects. Therefore, the feasibility of this needs to be investigated, possibly using multiple datasets with different objects. Alternative approaches for few-shot learning could also be investigated, involving additional training to expand the method, trained with abundant data for a few object classes, for the detection of new classes with just a few training samples, while maintaining performance in the original classes. The use of SAR simulations as templates, to detect new objects given only a 3-D model and without requiring the labelling of real SAR data, could also be investigated. Because the simulations will most likely not be perfectly realistic, a pseudo-Siamese network could also be used for this, giving the network more flexibility to learn how to compare real and simulated SAR data. Additionally, the advantages and disadvantages of the proposed network architecture with respect to a standard object detection network should be thoroughly analyzed, by applying both methods to the same datasets and comparing the obtained results. Some initial comparisons, which could unfortunately not be shown in this thesis, suggest that the proposed method compares favorably, but more experiments need to be performed.

Finally, as briefly illustrated in this thesis, the presented CD and ATR methods complement each other, and the combination and joint analysis of their results has shown a big potential. In this work, a simple strategy was used to combine their results, but a more sophisticated strategy should be developed, analyzing the results for the complete time series and combining them in a consistent manner. The combination of these two methods could also be applied for the monitoring other types of human activity, and should therefore be evaluated using datasets with different kinds of man-made objects in the future.

Bibliography

Abergel R, Denis L, Ladjal S, Tupin F (2018) Subpixellic Methods for Sidelobes Suppression and Strong Targets Extraction in Single Look Complex SAR Images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11 (3): 759–776.

Altshuler EE, Marr RA (1988) A comparison of experimental and theoretical values of atmospheric absorption at the longer millimeter wavelengths. IEEE Transactions on Antennas and Propagation, 36 (10): 1471–1480.

American Petroleum Institute (2013) API Standard 650: Welded Tanks for Oil Storage, 12 edition.

Anahara T, Shimada M (2018) Oil Storage Estimation with Time-Series L-Band Sar Imagery. In: 2018 IEEE International Geoscience & Remote Sensing Symposium: 842–845.

Anghelea A, Desnos YL, Maskey M, Sobue Si, Meissl S (2021) The COVID-19 Earth Observation Dashboard: A NASA-ESA-JAXA Collaborative Product. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS: 1556–1559.

Argenti F, Lapini A, Bianchi T, Alparone L (2013) A Tutorial on Speckle Reduction in Synthetic Aperture Radar Images. IEEE Geoscience and Remote Sensing Magazine, 1 (3): 6–35.

Aschbacher J (2017) ESA's earth observation strategy and Copernicus. In: Satellite earth observations and their impact on society and policy (pp. 81–86). Springer, Singapore.

Back M, Jeon T (2020) Analysis of Oil Storage Trend Using KOMPSAT-5 SAR Data. In: IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium: 4088–4091.

Bamler R, Hartl P (1998) Synthetic aperture radar interferometry. Inverse problems, 14 (4): R1.

Bao W, Hu J, Huang M, Xu Y, Ji N, Xiang X (2022) Detecting Fine-Grained Airplanes in SAR Images With Sparse Attention-Guided Pyramid and Class-Balanced Data Augmentation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15: 8586–8599.

Baumgartner SV, Krieger G (2016) Dual-Platform Large Along-Track Baseline GMTI. IEEE Transactions on Geoscience and Remote Sensing, 54 (3): 1554–1574.

Bazi Y, Bruzzone L, Melgani F (2006) Automatic Identification of the Number and Values of Decision Thresholds in the Log-Ratio Image for Change Detection in SAR Images. IEEE Geoscience and Remote Sensing Letters, 3 (3): 349–353.

Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PHS (2016) Fully-Convolutional Siamese Networks for Object Tracking. In: European conference on computer vision: 850–865.

Bovolo F, Bruzzone L (2015) The Time Variable in Data Fusion: A Change Detection Perspective. IEEE Geoscience and Remote Sensing Magazine, 3 (3): 8–26.

Bovolo F, Marin C, Bruzzone L (2013) A Hierarchical Approach to Change Detection in Very High Resolution SAR Images for Surveillance Applications. IEEE Transactions on Geoscience and Remote Sensing, 51 (4): 2042–2054.

Boyd DS, Jackson B, Wardlaw J, Foody GM, Marsh S, Bales K (2018) Slavery from Space: Demonstrating the role for satellite remote sensing to inform evidence-based action related to UN SDG number 8. ISPRS Journal of Photogrammetry and Remote Sensing, 142: 380–388.

Buckreuss S, Werninghaus R, Pitz W (2009) The German satellite mission TerraSAR-X. IEEE Aerospace and Electronic Systems Magazine, 24 (11): 4–9.

Butterworth RL (2004) Quill: The First Imaging Radar Satellite. United States National Reconnaissance Office.

Calabrese D, Torre A, Oddone A, Nicito A, Neglia SG, De Luca GF, Coletta A, Nirchio F, Libero CD (2018) COSMO-SkyMed mission status. In: EUSAR 2018; 12th European Conference on Synthetic Aperture Radar: 1–4.

Caltagirone F, Capuzi A, Coletta A, de Luca GF, Scorzafava E, Leonardi R, Rivola S, Fagioli S, Angino G, LAbbate M, Piemontese M, Zampolini Faustini E, Torre A, de Libero C, Esposito PG (2014) The COSMO-SkyMed Dual Use Earth Observation Program: Development, Qualification, and Results of the Commissioning of the Overall Constellation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7 (7): 2754–2762.

Castelletti D, Farquharson G, Stringham C, Duersch M, Eddy D (2021) Capella space first operational SAR satellite. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS: 1483–1486.

Cha M, Phillips RD, Wolfe PJ, Richmond CD (2015) Two-Stage Change Detection for Synthetic Aperture Radar. IEEE Transactions on Geoscience and Remote Sensing, 53 (12): 6547–6560.

Chen J, Yuan Z, Peng J, Chen L, Huang H, Zhu J, Liu Y, Li H (2021a) DASNet: Dual Attentive Fully Convolutional Siamese Networks for Change Detection in High-Resolution Satellite Images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 1194–1206.

Chen L, Luo R, Xing J, Li Z, Yuan Z, Cai X (2022) Geospatial Transformer Is What You Need for Aircraft Detection in SAR Imagery. IEEE Transactions on Geoscience and Remote Sensing, 60: 1–15.

Chen S, Wang H, Xu F, Jin YQ (2016) Target Classification Using the Deep Convolutional Networks for SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 54 (8): 4806–4817.

Chen S, Zhan R, Wang W, Zhang J (2021b) Learning Slimming SAR Ship Object Detector Through Network Pruning and Knowledge Distillation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 1267–1282.

Cong X, Balss U, Eineder M, Fritz T (2012) Imaging Geodesy—Centimeter-Level Ranging Accuracy With TerraSAR-X: An Update. IEEE Geoscience and Remote Sensing Letters, 9 (5): 948–952.

Crosetto M, Pérez Aragüés F (1999) Radargrammetry and SAR Interferometry for DEM Generation: Validation and Data Fusion. In: CEOS SAR Workshop, 450: 367.

Curlander JC (1982) Location of spaceborne SAR imagery. IEEE Transactions on Geoscience and Remote Sensing, (3): 359–364.

Cutrona L, Vivian W, Leith E, Hall G (1961) Synthetic aperture radars: A paradigm for technology evolution. IRE Trans. Military Electron, : 127–131.

Dalsasso E, Denis L, Tupin F (2022) As If by Magic: Self-Supervised Training of Deep Despeckling Networks With MERLIN. IEEE Transactions on Geoscience and Remote Sensing, 60: 1–13.

Deledalle CA, Denis L, Tupin F, Reigber A, Jager M (2015) NL-SAR: A Unified Nonlocal Framework for Resolution-Preserving (Pol)(In)SAR Denoising. IEEE Transactions on Geoscience and Remote Sensing, 53 (4): 2021–2038.

Deng, Zhang, Cai, Xu, Zhao, Guo, Suo (2019) Improvement and Assessment of the Absolute Positioning Accuracy of Chinese High-Resolution SAR Satellites. Remote Sensing, 11 (12): 1465.

Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition: 248–255.

Ding J, Xue N, Xia GS, Bai X, Yang W, Yang MY, Belongie S, Luo J, Datcu M, Pelillo M, Zhang L (2022) Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (11): 7778–7796.

Dong H, Ma W, Jiao L, Liu F, Li L (2022) A Multiscale Self-Attention Deep Clustering for Change Detection in SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 60: 1–16.

Du L, Li L, Di Wei, Mao J (2020) Saliency-Guided Single Shot Multibox Detector for Target Detection in SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 58 (5): 3366–3376.

Dudgeon D, Lacoss R (1993) An overview of automatic target recognition. The Lincoln Laboratory Journal, 6.

Eineder M, Adam N, Bamler R, Yague-Martinez N, Breit H (2009) Spaceborne Spotlight SAR Interferometry With TerraSAR-X. IEEE Transactions on Geoscience and Remote Sensing, 47 (5): 1524–1535.

Eineder M, Minet C, Steigenberger P, Cong X, Fritz T (2011) Imaging Geodesy—Toward Centimeter-Level Ranging Accuracy With TerraSAR-X. IEEE Transactions on Geoscience and Remote Sensing, 49 (2): 661–671.

Ester M, Kriegel HP, Sander J, Xu X (1996) Density-based spatial clustering of applications with noise. In: Int. Conf. Knowledge Discovery and Data Mining, 240: 6.

European Space Agency (2020) Earth observation for SDG: Compendium of earth observation contributions to the SDG targets and indicators.

Falkowski MJ, Maskey M, Sobue Si, Campbell G, Bawden G, Tadono T (2021) COIVD-19 Impact Monitoring of Economic Activities. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS: 1366–1369.

Fang S, Li K, Shao J, Li Z (2021) SNUNet-CD: A densely connected Siamese network for change detection of VHR images. IEEE Geoscience and Remote Sensing Letters, 19: 1–5.

Ferretti A, Prati C, Rocca F (2001) Permanent scatterers in SAR interferometry. IEEE Transactions on Geoscience and Remote Sensing, 39 (1): 8–20.

Gao F, Dong J, Li B, Xu Q, Xie C (2016) Change detection from synthetic aperture radar images based on neighborhood-based ratio and extreme learning machine. Journal of Applied Remote Sensing, 10 (4): 046019–046019.

Gao Y, Gao F, Dong J, Du Q, Li HC (2021) Synthetic Aperture Radar Image Change Detection via Siamese Adaptive Fusion Network. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 10748–10760.

Gatelli F, Monti Guamieri A, Parizzi F, Pasquali P, Prati C, Rocca F (1994) The wavenumber shift in SAR interferometry. IEEE Transactions on Geoscience and Remote Sensing, 32 (4): 855–865.

Ge N, Gonzalez FR, Wang Y, Shi Y, Zhu XX (2018) Spaceborne Staring Spotlight SAR Tomography—A First Demonstration With TerraSAR-X. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11 (10): 3743–3756.

Geng J, Ma X, Zhou X, Wang H (2019) Saliency-Guided Deep Neural Networks for SAR Image Change Detection. IEEE Transactions on Geoscience and Remote Sensing, 57 (10): 7365–7377.

Ghorbanzadeh O, Crivellari A, Tiede D, Ghamisi P, Lang S (2022) Mapping Dwellings in IDP/Refugee Settlements Using Deep Learning. Remote Sensing, 14 (24): 6382.

Giacovazzo VM, Refice A, Bovenga F, Veneziani N (07.07.2008 - 11.07.2008) Identification of Coherent Scatterers: Spectral Correlation vs. Multi-Chromatic Phase Analysis. In: IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium: IV – 411–IV – 414.

Gisinger C (2019) SAR Imaging Geodesy - Towards Absolute Coordinates with Centimeter Accuracy. Dissertation, Technical University of Munich, München.

Gisinger C, Balss U, Pail R, Zhu XX, Montazeri S, Gernhardt S, Eineder M (2015) Precise Three-Dimensional Stereo Localization of Corner Reflectors and Persistent Scatterers With TerraSAR-X. IEEE Transactions on Geoscience and Remote Sensing, 53 (4): 1782–1802.

Guida R, Iodice A, Riccio D (2010) Assessment of TerraSAR-X Products with a New Feature Extraction Application: Monitoring of Cylindrical Tanks. IEEE Transactions on Geoscience and Remote Sensing, 48 (2): 930–938.

Guo D, Wang J, Cui Y, Wang Z, Chen S (2020) SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition: 6269–6277.

Guo W, Yang W, Zhang H, Hua G (2018) Geospatial Object Detection in High Resolution Satellite Images Based on Multi-Scale Convolutional Neural Network. Remote Sensing, 10 (1): 131.

Hajnsek I, Desnos YL, eds (2021) Polarimetric Synthetic Aperture Radar: Principles and Application, volume 25 of Remote Sensing and Digital Image Processing. Cham: Springer International Publishing, 1st ed. 2021 edition.

Hamamoto K, Kuze A, Tadono T, Sobue S, Ishizawa J, Ohyoshi K, Murakami H, Kawamura K, Ikehata Y (2021) Jaxa's Earth Observation Data Analysis on Covid-19. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS: 1362–1365.

Hammer H, Kuny S, Schulz K (2017) Simulation-Based Signature Analysis of Fuel Storage Tanks in High-Resolution SAR Images. IEEE Geoscience and Remote Sensing Letters, 14 (8): 1278–1282.

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition: 770–778.

Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning: 448–456.

Jing M, Zhao D, Zhou M, Gao Y, Jiang Z, Shi Z (2019) Unsupervised Oil Tank Detection by Shape-Guide Saliency Model. IEEE Geoscience and Remote Sensing Letters, 16 (3): 477–481.

Jordan R (1980) The Seasat-A synthetic aperture radar system. IEEE Journal of Oceanic Engineering, 5 (2): 154–164.

Kang Y, Wang Z, Fu J, Sun X, Fu K (2022) SFR-Net: Scattering Feature Relation Network for Aircraft Detection in Complex SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 60: 1–17.

Kankaku Y, Suzuki S, Osawa Y (2013) ALOS-2 mission and development status. In: 2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS: 2396–2399.

Kim Dj, Hensley S, Yun SH, Neumann M (2016) Detection of Durable and Permanent Changes in Urban Areas Using Multitemporal Polarimetric UAVSAR Data. IEEE Geoscience and Remote Sensing Letters, 13 (2): 267–271.

Koch G, Zemel R, Salakhutdinov R et al. (2015) Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop, 2: 0.

Krieger G, Moreira A, Fiedler H, Hajnsek I, Werner M, Younis M, Zink M (2007) TanDEM-X: A Satellite Formation for High-Resolution SAR Interferometry. IEEE Transactions on Geoscience and Remote Sensing, 45 (11): 3317–3341.

Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. In: F. Pereira, C.J. Burges, L. Bottou, K.Q. Weinberger (eds) Advances in Neural Information Processing Systems, 25.

Kulu E (2021) Satellite Constellations - 2021 Industry Survey and Trends. In: 35th Annual Small Satellite Conference

Lafitte M, Robin J (2015) Monitoring nuclear facilities using satellite imagery and associated remote sensing techniques. ESARDA Bulletin, 52: 53–59.

Lee E, Jeong S, Kim J, Sohn K (2022) Semantic Equalization Learning for Semi-Supervised SAR Building Segmentation. IEEE Geoscience and Remote Sensing Letters, 19: 1–5.

Lei S, Lu D, Qiu X, Ding C (2021) SRSDD-v1. 0: A high-resolution SAR rotation ship detection dataset. Remote Sensing, 13 (24): 5104.

Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J (2019a) SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Li B, Yan J, Wu W, Zhu Z, Hu X (2018) High Performance Visual Tracking with Siamese Region Proposal Network. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition: 8971–8980.

Li J, Xu C, Su H, Gao L, Wang T (2022) Deep Learning for SAR Ship Detection: Past, Present and Future. Remote Sensing, 14 (11): 2712.

Li K, Wan G, Cheng G, Meng L, Han J (2020) Object detection in optical remote sensing images: A survey and a new benchmark. ISPRS journal of photogrammetry and remote sensing, 159: 296–307.

Li M, Li M, Zhang P, Wu Y, Song W, An L (2019b) SAR Image Change Detection Using PCANet Guided by Saliency Detection. IEEE Geoscience and Remote Sensing Letters, 16 (3): 402–406.

Li Y, Peng C, Chen Y, Jiao L, Zhou L, Shang R (2019c) A Deep Learning Method for Change Detection in Synthetic Aperture Radar Images. IEEE Transactions on Geoscience and Remote Sensing, 57 (8): 5751–5763.

Lin TY, Goyal P, Girshick R, He K, Dollar P (2017) Focal Loss for Dense Object Detection. In: 2017 IEEE International Conference on Computer Vision (ICCV): 2999–3007.

Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 11976–11986.

Liu Z, Zhao D, Shi Z, Jiang Z (2019) Unsupervised Saliency Model with Color Markov Chain for Oil Tank Detection. Remote Sensing, 11 (9): 1089.

Lobry S, Denis L, Tupin F (2016a) Multitemporal SAR Image Decomposition into Strong Scatterers, Background, and Speckle. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9 (8): 3419–3429.

Lobry S, Tupin F, Denis L (2016b) A decomposition model for scatterers change detection in multi-temporal series of SAR images. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS): 3362–3365.

Long Y, Xia GS, Li S, Yang W, Yang MY, Zhu XX, Zhang L, Li D (2021) On creating benchmark dataset for aerial image interpretation: Reviews, guidances, and million-aid. IEEE Journal of selected topics in applied earth observations and remote sensing, 14: 4205–4230.

Makhoul E, Baumgartner SV, Jager M, Broquetas A (2015) Multichannel SAR-GMTI in Maritime Scenarios With F-SAR and TerraSAR-X Sensors. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8 (11): 5052–5067.

Marin C, Bovolo F, Bruzzone L (2015) Building Change Detection in Multitemporal Very High Resolution SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 53 (5): 2664–2682.

Matar J, Rodriguez-Cassola M, Krieger G, Lopez-Dekker P, Moreira A (2020) MEO SAR: System Concepts and Analysis. IEEE Transactions on Geoscience and Remote Sensing, 58 (2): 1313–1324.

Mendez Dominguez E, Meier E, Small D, Schaepman ME, Bruzzone L, Henke D (2018) A Multisquint Framework for Change Detection in High-Resolution Multitemporal SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 56 (6): 3611–3623.

Meyer F, Hinz S, Laika A, Weihing D, Bamler R (2006) Performance analysis of the TerraSAR-X Traffic monitoring concept. ISPRS Journal of Photogrammetry and Remote Sensing, 61 (3-4): 225–242.

Mittermayer J, Wollstadt S, Prats-Iraola P, Scheiber R (2014) The TerraSAR-X Staring Spotlight Mode Concept. IEEE Transactions on Geoscience and Remote Sensing, 52 (6): 3695–3706.

Montazeri S (2019) Geodetic Synthetic Aperture Radar Interferometry. PhD thesis, Technical University of Munich.

Montazeri S, Gisinger C, Eineder M, Zhu XX (2018) Automatic Detection and Positioning of Ground Control Points Using TerraSAR-X Multiaspect Acquisitions. IEEE Transactions on Geoscience and Remote Sensing, 56 (5): 2613–2632.

Monti-Guarnieri AV, Brovelli MA, Manzoni M, Mariotti d'Alessandro M, Molinari ME, Oxoli D (2018) Coherent Change Detection for Multipass SAR. IEEE Transactions on Geoscience and Remote Sensing, 56 (11): 6811–6822.

Moreira A (2014) A golden age for spaceborne SAR systems. In: 2014 20th International Conference on Microwaves, Radar and Wireless Communications (MIKON): 1–4.

Moreira A, Prats-Iraola P, Younis M, Krieger G, Hajnsek I, Papathanassiou KP (2013) A tutorial on synthetic aperture radar. IEEE Geoscience and Remote Sensing Magazine, 1 (1): 6–43.

Muff D, Ignatenko V, Dogan O, Lamentowski L, Leprovost P, Nottingham M, Radius A, Seilonen T, Tolpekin V (2022) The ICEYE Constellation - Some New Achievements. In: 2022 IEEE Radar Conference (RadarConf22): 1–4.

Neubeck A, van Gool L (2006) Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR'06), 3: 850–855.

Ok AO, Baseski E (2015) Circular Oil Tank Detection From Panchromatic Satellite Images: A New Automated Approach. IEEE Geoscience and Remote Sensing Letters, 12 (6): 1347–1351.

Orbital Insight (n.d.) Orbital Insight Oil Inventories. `https://orbitalinsight.com/use-cases/oil-inventories`. (last accessed on February 5th, 2023).

Padilla R, Netto SL, Da Silva EA (2020) A survey on performance metrics for object-detection algorithms. In: 2020 international conference on systems, signals and image processing (IWSSIP): 237–242.

Palm SM (2021) Mapping of urban scenes by single-channel mmW FMCW SAR on circular flight and curved car trajectories. PhD thesis, Technical University of Munich.

Pan Z, Bao X, Zhang Y, Wang B, An Q, Lei B (2019) Siamese Network Based Metric Learning for SAR Target Classification. In: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium: 1342–1345.

Parizzi A, Cong X, Eineder M (2009) First Results from Multifrequency Interferometry. A comparison of different decorrelation time constants at L, C, and X Band. In: ESA FRINGE Workshop 2009: 1–5.

Persello C, Wegner JD, Hansch R, Tuia D, Ghamisi P, Koeva M, Camps-Valls G (2022) Deep Learning and Earth Observation to Support the Sustainable Development Goals: Current approaches, open challenges, and future opportunities. IEEE Geoscience and Remote Sensing Magazine, 10 (2): 172–200.

Preiss M, Gray DA, Stacy N (2006) Detecting scene changes using synthetic aperture Radar interferometry. IEEE Transactions on Geoscience and Remote Sensing, 44 (8): 2041–2054.

Pullarcot S (2015) Above Ground Storage Tanks: Practical Guide to Construction, Inspection, and Testing. CRC Press.

Reigber A, Jager M, Krogager E (2016) Polarimetric SAR change detection in multiple frequency bands for environmental monitoring in Arctic regions. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS): 5702–5705.

Reigber A, Moreira A, Papathanassiou KP (1999) First demonstration of airborne SAR tomography using multibaseline L-band data. In: IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99 (Cat. No.99CH36293): 44–46.

Rignot E, van Zyl JJ (1993) Change detection techniques for ERS-1 SAR data. IEEE Transactions on Geoscience and Remote Sensing, 31 (4): 896–906.

Romeiser R, Breit H, Eineder M, Runge H, Flament P, Jong Kd, Vogelzang J (2005) Current measurements by SAR along-track interferometry from a Space Shuttle. IEEE Transactions on Geoscience and Remote Sensing, 43 (10): 2315–2324.

Romeiser R, Suchandt S, Runge H, Steinbrecher U, Grunler S (2010) First Analysis of TerraSAR-X Along-Track InSAR-Derived Current Fields. IEEE Transactions on Geoscience and Remote Sensing, 48 (2): 820–829.

Rosen PA, Hensley S, Joughin IR, Li FK, Madsen SN, Rodriguez E, Goldstein RM (2000) Synthetic aperture radar interferometry. Proceedings of the IEEE, 88 (3): 333–382.

Ross TD, Worrell SW, Velten VJ, Mossing JC, Bryant ML (1998) Standard SAR ATR evaluation experiments using the MSTAR public release data set. In: Defense, Security, and Sensing

Sandia National Laboratories (n.d.) SAR Data - Pathfinder Radar ISR & SAR Systems. `https://www.sandia.gov/radar/pathfinder-radar-isr-and-synthetic-aperture-radar-sar-systems/complex-data/`. (last accessed on February 5th, 2023).

Sanjuan-Ferrer MJ (2013) Detection of coherent scatterers in SAR data: Algorithms and applications. PhD thesis, ETH Zurich.

Sanjuan-Ferrer MJ, Hajnsek I, Papathanassiou KP, Moreira A (2015) A New Detection Algorithm for Coherent Scatterers in SAR Data. IEEE Transactions on Geoscience and Remote Sensing, 53 (11): 6293–6307.

Sansosti E, Berardino P, Manunta M, Serafino F, Fornaro G (2006) Geometrical SAR image registration. IEEE Transactions on Geoscience and Remote Sensing, 44 (10): 2861–2870.

Schneider RZ, Papathanassiou K (12.07.2009 - 17.07.2009) Estimation and correction of ionospheric induced phase errors in SAR images using Coherent Scatterers. In: 2009 IEEE International Geoscience and Remote Sensing Symposium: IV–165–IV–168.

Schneider RZ, Papathanassiou KP, Hajnsek I, Moreira A (2006) Polarimetric and interferometric characterization of coherent scatterers in urban areas. IEEE Transactions on Geoscience and Remote Sensing, 44 (4): 971–984.

Schroff F, Kalenichenko D, Philbin J (2015) Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition: 815–823.

Shahzad M, Maurer M, Fraundorfer F, Wang Y, Zhu XX (2019) Buildings Detection in VHR SAR Images Using Fully Convolution Neural Networks. IEEE Transactions on Geoscience and Remote Sensing, 57 (2): 1100–1116.

Sheng H, Irvin J, Munukutla S, Zhang S, Cross C, Story K, Rustowicz R, Elsworth C, Yang Z, Omara M, Gautam R, Jackson RB, Ng AY (2020) OGNet: Towards a Global Oil and Gas Infrastructure Database using Deep Learning on Remotely Sensed Imagery.

Shi Y, Du L, Guo Y (2021) Unsupervised Domain Adaptation for SAR Target Detection. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 6372–6385.

Sica F, Gobbi G, Rizzoli P, Bruzzone L (2021) Φ-Net: Deep Residual Learning for InSAR Parameters Estimation. IEEE Transactions on Geoscience and Remote Sensing, 59 (5): 3917–3941.

Snoeij P, Attema E, Davidson M, Duesmann B, Floury N, Levrini G, Rommen B, Rosich B (2010) Sentinel-1 radar mission: Status and performance. IEEE Aerospace and Electronic Systems Magazine, 25 (8): 32–39.

Sprohnle K, Fuchs EM, Aravena Pelizari P (2017) Object-Based Analysis and Fusion of Optical and SAR Satellite Data for Dwelling Detection in Refugee Camps. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10 (5): 1780–1791.

Stankwitz H, Dallaire R, Fienup J (1995) Nonlinear apodization for sidelobe control in SAR imagery. IEEE Transactions on Aerospace and Electronic Systems, 31 (1): 267–279.

Steinbrecher U, Kraus T, Castellanos AG, Grigorov C, Schulze D, Braeutigam B (2014) TerraSAR-X: Design of the new operational WideScanSAR mode. In: EUSAR 2014; 10th European Conference on Synthetic Aperture Radar: 1–4.

Stringham C, Farquharson G, Castelletti D, Quist E, Riggi L, Eddy D, Soenen S (2019) The Capella X-band SAR Constellation for Rapid Imaging. In: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium: 9248–9251.

Su X, Deledalle CA, Tupin F, Sun H (2015) NORCAMA: Change analysis in SAR time series by likelihood ratio change matrix clustering. ISPRS Journal of Photogrammetry and Remote Sensing, 101: 247–261.

Suchandt S, Runge H, Breit H, Steinbrecher U, Kotenkov A, Balss U (2010) Automatic Extraction of Traffic Flows Using TerraSAR-X Along-Track Interferometry. IEEE Transactions on Geoscience and Remote Sensing, 48 (2): 807–819.

Sun X, Lv Y, Wang Z, Fu K (2022a) SCAN: Scattering Characteristics Analysis Network for Few-Shot Aircraft Classification in High-Resolution SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 60: 1–17.

Sun Y, Hua Y, Mou L, Zhu XX (2022b) CG-Net: Conditional GIS-Aware Network for Individual Building Segmentation in VHR SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 60: 1–15.

Sun Y, Mou L, Wang Y, Montazeri S, Zhu XX (2022c) Large-scale building height retrieval from single SAR imagery based on bounding box regression networks. ISPRS Journal of Photogrammetry and Remote Sensing, 184: 79–95.

Taigman Y, Yang M, Ranzato M, Wolf L (2014) DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition: 1701–1708.

Tao J, Auer S (2016) Simulation-Based Building Change Detection From Multiangle SAR Images and Digital Surface Models. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9 (8): 3777–3791.

Touzi R, Lopes A, Bruniquel J, Vachon P (1999) Coherence estimation for SAR imagery. IEEE Transactions on Geoscience and Remote Sensing, 37 (1): 135–149.

United Nations (2015) General Assembly Resolution A/RES/70/1. Transforming Our World, the 2030 Agenda for Sustainable Development.

United Nations Satellite Centre (2016) Damage density in the City of Aleppo, Syria. `https://unosat.org/products/1118`. (last accessed on February 5th, 2023).

United Nations Satellite Centre (2022) Kharkiv Damage Assessment Overview. `https://unosat.org/products/3455`. (last accessed on February 5th, 2023).

Ursa Space (n.d.) New Dataset: Oil Inventory Index. `https://ursaspace.com/blog/oil-inventory-index`. (last accessed on February 5th, 2023).

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems, 30.

Villamil Lopez C, Kempf T, Speck R, Anglberger H, Stilla U (2017) Automatic change detection using very high-resolution SAR images and prior knowledge about the scene. In: Ranney KI, Doerry A (eds) Radar Sensor Technology XXI: 1018805.

Villamil Lopez C, Stilla U (2018) Object-based SAR change detection for security and surveillance applications using density based clustering. In: EUSAR 2018; 12th European Conference on Synthetic Aperture Radar: 1–6.

Villamil Lopez C, Stilla U (2019) Using Coherent Scatterers in Time Series of High Resolution SAR Images for the Monitoring of Construction Activity. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W7: 183–187.

Villamil Lopez C, Stilla U (2021) Monitoring of Oil Tank Filling With Spaceborne SAR Using Coherent Scatterers. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 5638–5655.

Villamil Lopez C, Stilla U (2022) Monitoring of Construction Activity by Change Detection on SAR Time Series Using Coherent Scatterers. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15: 7499–7514.

Vladimir T, Vladimir V, Marina G, Leon N, Sergey Z, Evgeny T (2016) Russian Spaceborne Synthetic Aperture Radar Strizh for Light Satellites of Condor-E type. In: Proceedings of EUSAR 2016: 11th European Conference on Synthetic Aperture Radar: 1–6.

Wang D, Zhang Q, Xu Y, Zhang J, Du B, Tao D, Zhang L (2022) Advancing Plain Vision Transformer Towards Remote Sensing Foundation Model. IEEE Transactions on Geoscience and Remote Sensing, : 1–1.

Wang Q, Zhang J, Hu X (2016a) Automatic Oil Reserve Analysis Through the Shadows of Exterior Floating Crest Oil Tanks in Highlight Optical Satellite Images. In: Bebis G, Boyle R, Parvin B, Koracin D, Porikli F, Skaff S, Entezari A, Min J, Iwai D, Sadagic A, Scheidegger C, Isenberg T (eds) Advances in Visual Computing, volume 10073 of Lecture Notes in Computer Science (pp. 23–32). Cham: Springer International Publishing.

Wang Q, Zhang J, Hu X, Wang Y (2016b) Automatic Detection and Classification of Oil Tanks in Optical Satellite Images Based on Convolutional Neural Network. In: Mansouri A, Nouboud F, Chalifour A, Mammass D, Meunier J, Elmoataz A (eds) Image and Signal Processing, volume 9680 of Lecture Notes in Computer Science (pp. 304–313). Cham: Springer International Publishing.

Wang T, Li Y, Yu S, Liu Y (2019) Estimating the Volume of Oil Tanks Based on High-Resolution Remote Sensing Images. Remote Sensing, 11 (7): 793.

Wei S, Zeng X, Qu Q, Wang M, Su H, Shi J (2020) HRSID: A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation. IEEE Access, 8: 120234–120254.

Wei S, Zeng X, Zhang H, Zhou Z, Shi J, Zhang X (2022) LFG-Net: Low-Level Feature Guided Network for Precise Ship Instance Segmentation in SAR Images. IEEE Transactions on Geoscience and Remote Sensing, 60: 1–17.

Wiley CA (1954) Pulsed Doppler radar methods and apparatus. US. Patent, 3.

Willey CA (1985) Synthetic aperture radars–a paradigm for technology evolution. IEEE Trans. Aerospace Elec. Sys, 21: 440–443.

Wu C, Zhu S, Yang J, Hu M, Du B, Zhang L, Zhang L, Han C, Lan M (2021) Traffic Density Reduction Caused by City Lockdowns Across the World During the COVID-19 Epidemic: From the View of High-Resolution Remote Sensing Imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 5180–5193.

Wu Q, Zhang B, Xu C, Zhang H, Wang C (2022) Dense Oil Tank Detection and Classification via YOLOX-TR Network in Large-Scale SAR Images. Remote Sensing, 14 (14): 3246.

Wulder MA, Loveland TR, Roy DP, Crawford CJ, Masek JG, Woodcock CE, Allen RG, Anderson MC, Belward AS, Cohen WB, Dwyer J, Erb A, Gao F, Griffiths P, Helder D, Hermosilla T, Hipple JD, Hostert P, Hughes MJ, Huntington J, Johnson DM, Kennedy R, Kilic A, Li Z, Lymburner L, McCorkel J, Pahlevan N, Scambos TA, Schaaf C, Schott JR, Sheng Y, Storey J, Vermote E, Vogelmann J, White JC, Wynne RH, Zhu Z (2019) Current status of Landsat program, science, and applications. Remote Sensing of Environment, 225: 127–147.

Xia R, Chen J, Huang Z, Wan H, Wu B, Sun L, Yao B, Xiang H, Xing M (2022) CRTransSar: A Visual Transformer Based on Contextual Joint Representation Learning for SAR Ship Detection. Remote Sensing, 14 (6): 1488.

Xu H, Chen W, Sun B, Chen Y, Li C (2014) Oil tank detection in synthetic aperture radar images based on quasi-circular shadow and highlighting arcs. Journal of Applied Remote Sensing, 8 (1): 083689.

Xu S, Zhang H, He X, Cao X, Hu J (2022) Oil Tank Detection With Improved EfficientDet Model. IEEE Geoscience and Remote Sensing Letters, 19: 1–5.

Xu Y, Wang Z, Li Z, Yuan Y, Yu G (2020) SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines. In: Proceedings of the AAAI Conference on Artificial Intelligence, 34: 12549–12556.

Yang CH, Soergel U (2018) Adaptive 4D Change Detection Based on PSI. In: EUSAR 2018; 12th European Conference on Synthetic Aperture Radar: 1–5.

Yu J, Wang Z, Majumdar A, Rajagopal R (2018) DeepSolar: A Machine Learning Framework to Efficiently Construct a Solar Deployment Database in the United States. Joule, 2 (12): 2605–2617.

Zhang L, He X, Balz T, Wei X, Liao M (2011) Rational function modeling for spaceborne SAR datasets. ISPRS Journal of Photogrammetry and Remote Sensing, 66 (1): 133–145.

Zhang L, Liu C (2020) Oil Tank Extraction Based on Joint-Spatial Saliency Analysis for Multiple SAR Images. IEEE Geoscience and Remote Sensing Letters, 17 (6): 998–1002.

Zhang L, Lu Z (2022) Advances in InSAR Imaging and Data Processing. Remote Sensing, 14 (17): 4307.

Zhang L, Shi Z, Wu J (2015) A Hierarchical Oil Tank Detector With Deep Surrounding Features for High-Resolution Optical Satellite Imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8 (10): 4895–4909.

Zhang L, Wang S, Liu C, Wang Y (2019) Saliency-Driven Oil Tank Detection Based on Multidimensional Feature Vector Clustering for SAR Images. IEEE Geoscience and Remote Sensing Letters, 16 (4): 653–657.

Zhang P, Xu H, Tian T, Gao P, Li L, Zhao T, Zhang N, Tian J (2022) SEFEPNet: Scale Expansion and Feature Enhancement Pyramid Network for SAR Aircraft Detection With Small Sample Dataset. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15: 3365–3375.

Zhang T, Zhang X, Li J, Xu X, Wang B, Zhan X, Xu Y, Ke X, Zeng T, Su H, Ahmad I, Pan D, Liu C, Zhou Y, Shi J, Wei S (2021) SAR Ship Detection Dataset (SSDD): Official Release and Comprehensive Data Analysis. Remote Sensing, 13 (18): 3690.

Zhao Y, Zhao L, Liu Z, Hu D, Kuang G, Liu L (2022) Attentional Feature Refinement and Alignment Network for Aircraft Detection in SAR Imagery. IEEE Transactions on Geoscience and Remote Sensing, 60: 1–16.

Zhou Y, Jiang X, Xu G, Yang X, Liu X, Li Z (2023) PVT-SAR: An Arbitrarily Oriented SAR Ship Detector With Pyramid Vision Transformer. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 16: 291–305.

Zhu S, Song Y, Zhang Y, Zhang Y (2023) ECFNet: A Siamese Network with Fewer FPs and Fewer FNs for Change Detection of Remote Sensing Images. IEEE Geoscience and Remote Sensing Letters.

Zhu X (2011) Very High Resolution Tomographic SAR Inversion for Urban Infrastructure Monitoring: A Sparse and Nonlinear Tour. PhD thesis, Technical University of Munich.

Zhu X, Wang Y, Montazeri S, Ge N (2018) A Review of Ten-Year Advances of Multi-Baseline SAR Interferometry Using TerraSAR-X Data. Remote Sensing, 10 (9): 1374.

Zhu XX, Montazeri S, Ali M, Hua Y, Wang Y, Mou L, Shi Y, Xu F, Bamler R (2021) Deep Learning Meets SAR: Concepts, models, pitfalls, and perspectives. IEEE Geoscience and Remote Sensing Magazine, 9 (4): 143–172.

Zhu XX, Tuia D, Mou L, Xia GS, Zhang L, Xu F, Fraundorfer F (2017) Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. IEEE Geoscience and Remote Sensing Magazine, 5 (4): 8–36.

Zink M, Moreira A, Hajnsek I, Rizzoli P, Bachmann M, Kahle R, Fritz T, Huber M, Krieger G, Lachaise M, Martone M, Maurer E, Wessel B (2021) TanDEM-X: 10 Years of Formation Flying Bistatic SAR Interferometry. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 3546–3565.

Zou B, Qin J, Zhang L (2022) Vehicle Detection Based on Semantic-Context Enhancement for High-Resolution SAR Images in Complex Background. IEEE Geoscience and Remote Sensing Letters, 19: 1–5.

Acknowledgment

First of all, I would like to express my deepest gratitude to my doctoral supervisor, Prof. Uwe Stilla, who allowed me to pursue a Ph.D. degree at the Technical University of Munich. Over the course of the Ph.D. I benefitted greatly from our many discussions, where he always provided insightful feedback and constructive criticism, helping me to improve the quality of my work. In addition to supervising my research and helping me to structure this dissertation, he also provided valuable advice regarding the process of scientific publishing and taught me how to write in a concise and precise manner. And even while sick, he still took the time to support me and help me to finish and submit this dissertation. For this, I will always be thankful to him.

Many thanks to Prof. Alberto Moreira, who, besides serving as the third reviewer of this thesis, has also supported my Ph.D. work throughout the years in his role as director of DLR's Microwaves and Radar Institute. Moreover, I would also like to thank Prof. Martin Werner and Prof. Michael Schmitt for their work as the first and second reviewers of this thesis, and Prof. Niklas Boers for serving as the chairman of the examination committee.

I had the pleasure of writing this dissertation alongside my work as a research scientist at DLR's Microwaves and Radar Institute. For me, this has been the best possible place for learning about spaceborne SAR, thanks to all the colleagues who have shared their expertise with me during seminar presentations, project meetings, or informal talks. I am extremely grateful to my supervisors at DLR, Dr. Rainer Speck and Dr. Harald Anglberger, for mentoring and supporting me before and during the Ph.D., and for continuing to do so even now after it is over. Special thanks go to my colleagues at DLR's SAR Simulation research group: Dr. Timo Kempf, Manfred Hager, Ismail Baris, Juliane Profelt, and Dr. Andreas Heinzel, without you this dissertation would have not been possible. I also want to thank the rest of my colleagues at the Reconnaissance and Security department, especially Dr. Markus Peichl and Stephan Dill, who gave me the opportunity to come to Germany and start working at DLR; as well as Bettina Thurner and Gabriele Hager, who provided the best possible support in all administrative matters. I would also like to acknowledge everyone involved in the operation of the TerraSAR-X and TanDEM-X satellites, which provided the best possible data for my work. Furthermore, thanks to everyone involved in the Kephale project, many of whom contributed in some way to the results on airplane recognition. Here, I want to mention Lars Petersen in particular, who played a vital role in the creation of the airplane dataset; and Dr. Ronny Hänsch, Dr. Francescopaolo Sica, and Andrea Pulella, for all the discussions and their suggestions on the design of the experiments. Finally, big thanks to my colleagues at TerraLens GmbH for your patience during the final phase of my Ph.D.: you allowed me to focus on writing this thesis, even when that meant that I had to skip some of our meetings or I did not come to the office for weeks at a time.

Last, but certainly not least, I want to thank my friends and family. To my friends in Munich, thank you for being like a second family, and for all those after-work beers, paellas, and weekends spent traveling, skiing, or hiking; you made even the stressful times fun. To my brother, thank you for always inspiring and challenging me: you taught me Math as a little kid, later encouraged me to travel and move abroad, and have been providing valuable advice ever since I decided to start a company. Finally, the most significant acknowledgment is reserved for my parents: thank you for literally everything. It is only thanks to your continuous and unconditional support that I was able to achieve everything that I have.