UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CORSO DI LAUREA MAGISTRALE IN
GESTIONE DEI SISTEMI AEROSPAZIALI PER LA DIFESA

Sentinel-1 and -2 Data Fusion through Generative Adversarial Network for Image Simulation

Tesi di Laurea Magistrale in
Space Systems

Relatore:
Ch.mo Prof. Antonio MOCCIA

Candidato:
Matteo INGROSSO
Matr. P24/37

Anno Accademico 2019/2020

Abstract

The growing interest in dual-use satellite systems and the recent advancements in machine learning open new potential remote sensing applications. Data fusion is also a task of interest since it can produce high-value products. This work moves from similar research and investigates the implementation of deep learning algorithms to accomplish a data fusion task on Sentinel imagery. The objective is to simulate an optical image using multi-temporal SAR-optical images. The tests aimed to prove the functionality of the chosen method across different datasets and how performance can change with the quantity and quality of training. The method used successfully created a simulated image that resembles its original counterpart, but the result is still blurred and relatively effective. The tests prove that hardware limitations can affect the method's potential since larger images and deep networks need computational power. Moreover, the training behavior of GANs is still an open problem and needs to be stabilized to achieve better performance. Finally, the dataset quality is essential and can affect the performance both negatively and positively.

Contents

List of Figures

List of Tables

Introduction

This thesis work is mainly focused on deep learning techniques applied to satellite imagery to accomplish a data fusion task for image simulation. The practical application is supported by theoretical information. So, this thesis is divided into three chapters.

In the first chapter, there is a general explanation of the Copernicus program, with an introduction to its Earth Observation (EO) capabilities and mission. One paragraph focuses its attention on the market outcome of Copernicus, with a discussion on the changing trends related to brand-new technologies. Then, the infrastructure is explained, with the main focus on the space segment and the two satellite systems used for the project (Sentinel-1 and -2). Finally, the services that Copernicus makes available are shown.

The second chapter is related to the theoretical side of the project, and it focuses on the software part. It explains what the Sentinel Application Platform (SNAP) is, and the features that were used. Then, there is a broad explanation of what machine learning is, with an overview of the primary methods used in the project.

Finally, the third chapter covers the experimental part, starting from the problem formulation, explaining the adopted methodology with proper references and showing the results achieved. After the conclusion, a paragraph discusses the potential applications of this project and related ideas for the needs of the Defense Department.

Chapter 1

The Copernicus program

> *At rest, however, in the middle of everything is the sun. [...] Thus indeed, as though seated on a royal throne, the sun governs the family of planets revolving around it.*
>
> — **Nicolaus Copernicus**

Copernicus is the European Union's Earth Observation and Monitoring program. Previously known as Global Monitoring for Environment and Security (GMES), it was named after Copernicus in 2012 and aims to establish a European capacity for Earth Observation (EO). The European Commission is the leading actor and is in charge of setting the political vision of the program and supporting the proper functioning of the system; then, together with the European Parliament and Council, it ensures the sustainability of the program by planning the long-term financial commitment. In practice, the European Commission is responsible for the implementation of the program through its three components, that are discussed thoroughly later in this work:

- Space component
- In-situ component
- Copernicus services component

The partnership extends to the European Union (EU) Member States and European Space Agency (ESA) for a deep collaboration system through the development of satellites and the sharing of data and infrastructures. ESA is also responsible for the Space Segment and its technical coordination; satellites' operations have been entrusted to ESA and EUMETSAT, while they are a property

of the EU. All the services that are provided by the program are performed either by the Joint Research Centre (JRC) or an appropriate entity or agency: in this set, we found the European Environment Agency (EEA), FRONTEX (European Agency for the Management of Operational Cooperation at the External Borders of the Member States of the European Union, from French *Frontières extérieures*), European Maritime Safety Agency (EMSA), European Satellite Centre (SatCen), European Centre for Medium-Range Weather Forecasts (ECMWF) and Mercator Océan [1].

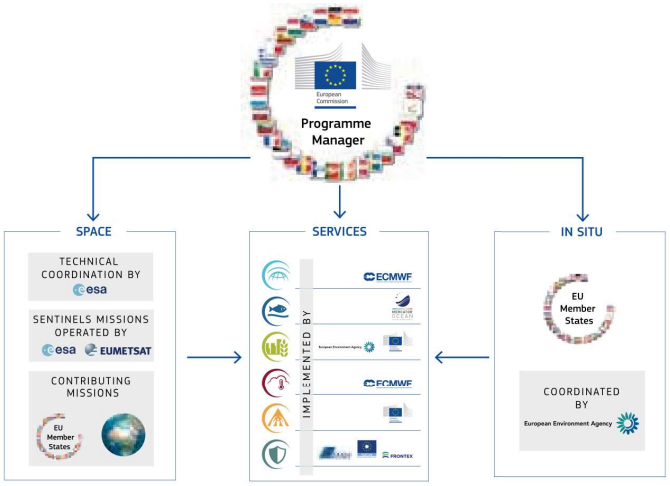


Figure 1.1: Components with related actors of the program [1].

1.1 Earth Observation

EO is the activity that involves the use of remote sensing technologies to monitor the Earth's surface and atmosphere. The satellites built for this reason use their payload to gather images and data about Earth's features. Then, during the processing phase, images and data are merged and analyzed to obtain different types of information that can be exploited by several users and applications.

The kinds of sensors used in EO split into two different families:

- **Optical and thermal sensors**: they sense natural energy from the Sun reflected or re-emitted by the Earth's surface or atmosphere; they work in the visible and IR wavelengths;
- **Radar sensors**: they are active sensors that send radiation (in this case with longer wavelengths) and then sense the reflected echo from the surface; they enable all-day and all-weather monitoring.

In the activity of EO, the resolution plays a pivotal role. It is a parameter that estimates the capability to distinguish different targets:

- **Spatial** resolution determines the size of the pixels analyzed by the sensors, i.e. how vast is the area compressed in one pixel; the satellites are distributed into categories according to how many meters there are per unit of image, so they vary in range from more than 10 m (low or medium resolution) to sub-metric values (very high resolution);
- **Temporal** resolution is the frequency at which the data is acquired for a defined area, and it varies depending on the needs;
- **Spectral** resolution is the width of the bands that can be discriminated by the payload, enabling the selection of specific bands or a range of them.

The list of essential parameters extends to the coverage and the revisit frequency. Satellites allow a global coverage with a single spacecraft because it flies in orbit, while in-situ sensors maintain the local coverage. On the other hand, orbital mechanics affect the revisit time; thus, a spacecraft flies over the same location after some time (that is usually once per day or few days), while land monitoring is more persistent.

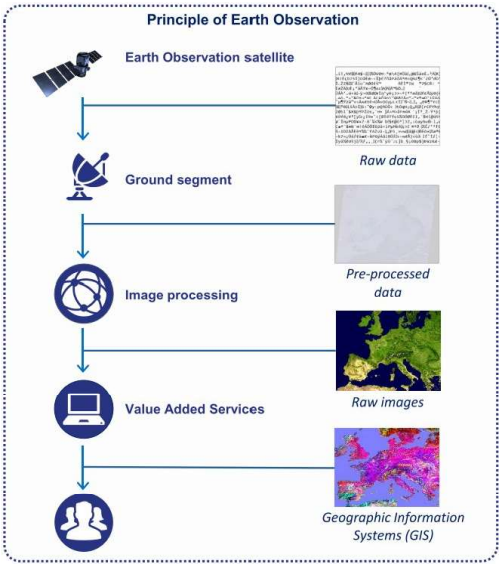


Figure 1.2: EO workflow [1].

1.2 Mission of Copernicus

According to Elżbieta Bieńkowska (European Commissioner for Internal Market, Industry, Entrepreneurship and Small and Medium Entrepreneurs), "Copernicus will deliver an unprecedented volume of free data, provide new operational services and foster new business opportunities and job creation." A comprehensive definition may be that the program is designed to support the goals regarding sustainable development and the global governance of the environment by providing timely and quality data, information, services and knowledge [2].

In this way, it is a brand-new approach to Earth observation that is likely to set a new standard in the field and represents a milestone in the European history of space exploration. It builds on ample know-how to develop a system that can last longer and be extremely useful for the whole planet. It starts with the simple observation and data collection through extensive and straightforward infrastructure; data are then stored and analyzed to provide several different products for the end-users. Hence, these outputs can be merged with each other and with data from international systems to create a reliable and up-to-date network of information for statistics and to build indicators and temporal data.

Furthermore, a specific trait of Copernicus is that it is entirely open and free: anyone, from scientists and policymakers to entrepreneurs and citizens, has full and open access to data free of charge. Thus, it creates a public service framework that can increase the outcome of the system exponentially: delivering data to everyone allows people to work with Copernicus' information and to develop new applications for their specific needs. It is meant to boost scientific research, to support political and meaningful decisions, and to enhance our understanding of planet changes.

The reason behind Copernicus grows with the idea of tackling modern environmental and human-related challenges: nowadays and soon, we are facing several global threats, from climate changes, energy crisis, overpopulation, potential food shortage and recurrent yet impressive natural disasters. Along these lines, Copernicus wants to ease the management of these scenarios and boost the overall attention towards societal goals. So, the output of the value chain will help to make progress in renewable energy sources, food security, risk reduction related to disasters and climate change mitigation. As a result, our awareness of the planet status can be improved to allow more rational policymaking [1].

As a civil and European program, Copernicus is open to the needs of its users and has a twofold service towards European citizens: directly through products

and applications, and indirectly through economic and environmental benefits. It is a point excellence of the European effort in the field of the space industry and, thus, a highly strategic system.

In this way, the outcome of Copernicus fosters the EU's role on the international stage and advances its position as a global actor. Thereby, the European role in space is strengthened. As a single entity, the EU requires independent information to access both bilateral and multilateral initiatives and to support its position during negotiations on climate, environment and biodiversity. Indeed, these current topics have paramount importance in the current and future European policy. However, although it is a European project, it is not restricted to European citizens; since the access to data is full, free and open to everyone, the whole international community will benefit from Copernicus in responding and adapting to global phenomena.

The program is also a joining venture of ESA and all the member countries to back a continuous, sustainable and high-level system as a result of a long-term European commitment. Copernicus builds on strong national capacities and expertise grown in many years of research and development; it was also useful to optimize the costs related to long-duration EO activities.

From the idea of tackling modern environmental and human-related challenges, Copernicus has been growing to become a powerful, interconnected, extensive and limitless system with a high return on investment.

1.2.1 Market

Since the space sector is growing at a fast speed and has become pivotal in modern society and economy, Copernicus serves as stimulation for European enterprises to look for new business opportunities and job creation. The market that the program generates is getting more extensive as its applications advance [1].

As stated in [3], in the timeframe of 2008-2020, Copernicus is forecasted to reach €8.2 billion as total investments. In contrast, the economic benefits over the same period are estimated between €16.2 and €21.3 billion, excluding non-monetary benefits: as an example, reduced casualties during natural disasters, increased food security and improved air quality. The economic value is generated thanks to the upstream space industry, the sales of Copernicus-based applications by downstream service suppliers and the exploitation of Copernicus-enabled products by end-users.

So, companies are involved in the activities by the three main branches of EO activity:

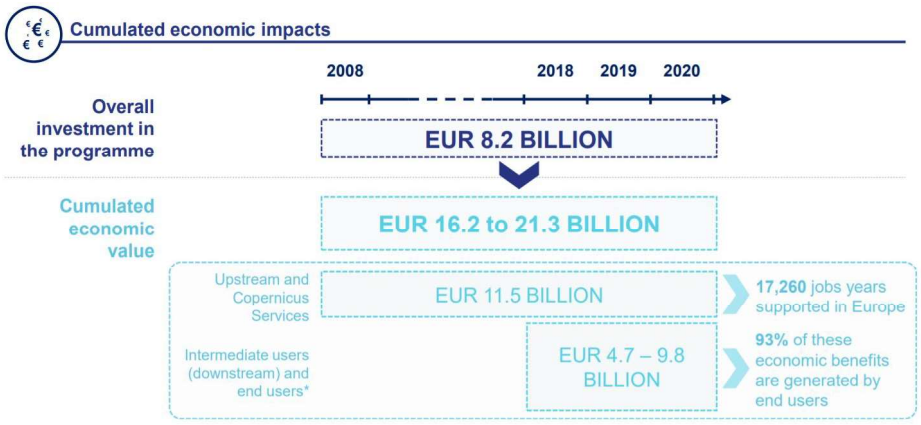


Figure 1.3: Overall economic impact of Copernicus [1].

- The **upstream industry** entails the space industry responsible for the development and manufacture of the infrastructure, i.e. space and ground segments for satellite operations, including the launch operations;
- The **downstream industry** includes all the actors and companies whose activities orbit around the processing of data to create Value Added Services (VAS); they are meant as intermediate users and are experts in EO;
- The **end users** are those whose core business is not related to EO, so the derived products made by the intermediate users are inputs for them; they are institutional players, entrepreneurs and people that need ready and qualified information.

So far, it is possible to see how Copernicus extends more beyond its space missions until involving anyone who needs modern and reliable solutions based on EO. The role of the intermediate users is focal since the value of raw data and images is given by the applications derived from them.

Moreover, applications grow with technology. End users provide an essential service by linking VASes with their relevant sources of information. In this way, data integration increases exponentially and adapts to businesses; in the frame of combining multiple digital sources, Big Data is expanding the possibilities even more. So, people are updating procedures at every level and are tailoring their products to offer better products and to fit day-to-day challenges.

1.2.2 Changing trends

The characteristics of such a program are changing and evolving as technology and applications grow.

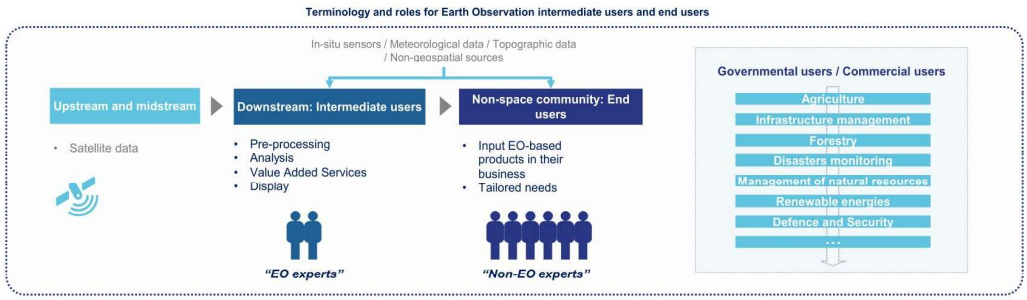


Figure 1.4: Roles along the value chain with users difference [1].

New users are being involved and targeted in using EO satellites and data. Two areas are demanding rapid improvement as they expand: they are emerging economies, where brand-new solutions can support the growth of countries, and the intelligence community for security and defense. Then, the decision-making requirements of non-technical users boost the delivery of tailored and focused insights that would not have been needed previously.

EO companies are upgrading their business models to match the emergence of low-cost small satellites. They provide near real-time, low-resolution imagery more cheaply, thanks to Commercial Off-the-Shelf (COTS) components. Since the dimension diminishes, it is possible to put large constellations in orbit, increasing the amount of data. Then, data providers are adapting to the shift of the market, so they want to meet the need for technical expertise in the growing community of end-users.

Another technology that advances the access to data is cloud computing. It eases the storage of large volumes of data and reduces the cost, giving users a wide range of sources in a single point. Moreover, cloud computing is the on-demand delivery of computer power, servers, databases, networking, software and analytics that together increase the possibilities of modern solutions. In Copernicus, for instance, a network of servers with large repositories supports the interface between users and data (e.g., Copernicus hub and Long-Term Archives). Then, ESA provides Sentinel Application Platform (SNAP), which is a software to process and analyze data from Sentinel satellites.

Moreover, brand-new solutions in data handling and analytics, together with the integration of different types of data, have led to the adoption of artificial intelligence, machine learning algorithms, data mining and data fusion techniques to derive additional insights and build a comprehensive awareness in complex situations [3]. It is worth saying that the concept of Space Situational Awareness has gained a pivotal role in the Space Strategy for Europe [4].

EO solutions usage is also affected by users' willingness to pay and the availability of data. Copernicus has made data freely accessible to any user, following the shift towards data democratization. However, some end-users need high-resolution and real-time, as well as high reliability, and for this reason, they are willing to pay more than those who have lower requirements. For instance, there is a considerable gap between the intelligence and the agriculture sectors, both services covered by Copernicus [1].

Also, Big Data solutions need modern circumstances to give a reason to such large internal repositories of data. They are large volumes of data too complex for traditional processing, but with great importance, because they can reveal trends and connections. Big Data Analytics requires modern resources to process, analyze and fuse multiple images and other data sources to create new intelligence. In this fast-growing market, Copernicus is one of the key actors thanks to its continuity and volume of data.

Overall, users that need flexibility and responsiveness are the defense and emergency actors. The availability of such a program should not create failures of national security, so people working in this field need cutting-edge technology and high-tech intelligence to tackle cases that range from border surveillance and crisis monitoring to civil disturbance and terrorism [3].

1.3 Infrastructure

The Copernicus infrastructure is the set of systems that provide data collection. Data are collected via the space and in-situ components, two main directories of the whole program.

1.3.1 Space component

The space component encloses EO satellites that gather data from orbit, and it is categorized into two groups:

- The family of satellites dedicated to Copernicus The Sentinels;
- A set of nearly 30 contributing missions from various organizations.

It also includes the procurement and launch operations of the Sentinels, the ground segment and the distribution of data.

The Sentinels are EU-owned satellites and are technically managed by ESA. They are specifically designed to meet the needs of the program [3]. Sentinel-1, -2, -3, -5P and -6 are satellites with different sensors, while Sentinel-4 and -5 are instruments on-board weather satellites belonging to EUMETSAT [3]. They include radar and multispectral imaging for land, maritime and atmospheric monitoring. Each mission is made up of two satellites to meet the revisit and coverage requirements.

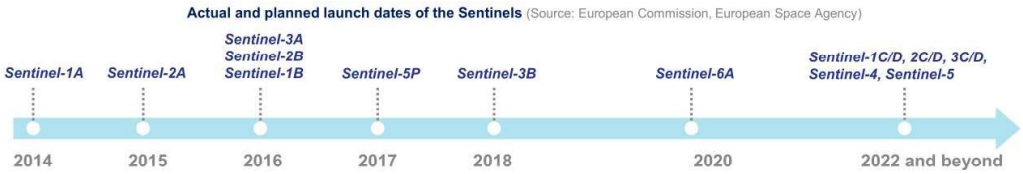


Figure 1.5: Chronological succession of Sentinel missions [1].

ESA is responsible for the development of the space segment and operates Sentinel-1, -2, -5P, whereas it delivers only the land mission for Sentinel-3. EUMETSAT is responsible for Sentinel-3 and -6 operations and delivers products from the Sentinel on-board its missions. The two agencies also coordinate for the delivery of data from the contributing missions. As a completion, a ground segment exists to allow Sentinels to work correctly, through the mission control, and to provide access to data; the segment is spread geographically and relies on existing infrastructures of European agencies and national property.

In addition, the contributing missions have a pivotal role in delivering complementary high-resolution data that would not be obtained using Sentinels. They exist independently and are operated by European agencies as well as third party countries and commercial providers, depending on the property of the system. They cover different features, as follows [1], [3], [5]:

- Synthetic Aperture Radar (SAR) to observe day and night the land and the ocean;
- Optical sensors to monitor land activities and ocean dynamics;
- Altimetry systems for sea-level measurement;
- Radiometers to monitor land and ocean temperature;
- Spectrometers to measure air quality.

A specific description of only Sentinel-1 and -2 satellite systems follows because they play a crucial role in the application of this thesis; however, more Sentinel missions exist for different purposes.

Sentinel-1

In the frame of Copernicus, Sentinel-1 was thought to develop the European Radar Observatory. It consists of a two-satellite constellation, placed in polar orbit, for the continuation and improvement of SAR services and to meet the revisit and coverage requirements. It aims to provide routine, day-and-night, all-weather medium resolution (down to 10 m) observation capability. Sentinel-1 was built on the heritage of past Radar and SAR missions, such as European ERS or Advanced SAR on Envisat. The satellites' lifetime is set at 7 years, with consumables on-board which allow a mission extension up to 12 years. The life cycle of a satellite generation is planned in the order of 15-20 years [6].

Sentinel-1 works in a pre-programmed conflict-free operation mode, imaging landmasses, coastal zones, sea-ice, polar areas and shipping routes at high resolution. In this way, it ensures reliability of service and a persistent long-term data archive. In the framework of international agreements, the revisit and coverage performance may be further improved with the support of Canadian C-band SAR constellation [6].

The mission design is determined by a set of observation requirements, such as data availability, coverage and revisit, timeliness and characteristics of data products.

A conventional SAR system combines different operational modes that can optimize either the spatial resolution or the swath width. So, Sentinel-1 was meant to focus on medium- to high-resolution applications via a primary operational mode, which features a wide swath (250km) and high geometric (5m x 20m) and radiometric resolution. For the needs of some service providers, the extra-wide swath mode may be used over sea-ice and polar zones to ensure broader coverage and better revisit time at the expense of the resolution [6].

The imaging modes can be operated for a maximum of 25 minutes per orbit. The remaining time the system operates over the ocean in Wave Mode, delivering sample images of 20x20 km at 100 km with a low data rate.

Sentinel-1 characteristics are in table 1.1. The satellite has been built by a consortium led by Thales Alenia Space Italy as prime contractor, with Astrium Germany responsible for the C-band SAR (CSAR) payload and the central radar electronics developed by Astrium UK. The spacecraft is built on the Piattaforma Italiana Multi Applicativa (PRIMA) platform, featuring a CSAR instrument in this case. The expertise was developed from the Canadian RadarSat-2 and

Table 1.1: Sentinel-1 spacecraft characteristics [6].

Lifetime	7 years (consumables for 12 years)
Orbit	Near Polar Sun-Synchronous @ 693 km; 12-day repeat cycle; 175 orbits per cycle
Mean local solar time	18:00 at Ascending Node
Orbital period	98.6 min
Max eclipse duration	19 min
Attitude stabilisation	3 axis stabilised
Attitude accuracy	0.01° (each axis)
Nominal flight attitude	Right looking
Steering	Zero Doppler yaw steering and roll steering (−0.8 to 0.8°)
Attitude profile	Geo-centric and geodetic
Orbit knowledge	10 m (each axis, 3 sigma) using GPS
Operative autonomy	96 h
Launch mass	2300 kg (incl. 130 kg monopropellant fuel)
Dimensions (stowed)	3900×2600×2500 mm^3
Solar array average power	5900 W (End-of-Life)
Battery capacity	324 Ah
Spacecraft availability	0.998
S-Band Telemetry, Tracking and Command (TT&C) data rates	64 kbps Telecommand; 128 kpbs/2 Mbps Telemetry (programmable)
Launcher	Soyuz from Kourou

Cosmo-Skymed programs that used PRIMA as well.

The spacecraft built for this mission was placed into Earth orbit directly by the launcher and weighted 2300 kg at launch. It is a 3-axis stabilized satellite and features Sun, star, gyro and magnetic field sensors. A set of 4 reaction wheels is dedicated to orbit and attitude control; steering capabilities on each axis are provided by 3 torque rods as actuators. The satellite is equipped with 2 solar array wings that produce 5900 W at the end of life to be stored in a modular battery [6].

The platform provides excellent performance: on each axis, it has accurate pointing knowledge (better than 0.004°) and high pointing accuracy (about 0.01°), together with real-time orbit determination. A dedicated propulsion system grants precise orbit control. The reference orbit is maintained within an Earth-fixed orbital tube with a diameter of 100 m during nominal operation time. The two satellites, Sentinel-1A and -1B, have a 180° orbital phasing difference, so they have the same reference every half orbit.

The backbone of the system is based on the MIL-STD-1553 command bus with the ERC32-SC processor; together, they provide high computing performance, fast avionics and reliable satellite communication [6].

Regarding the modes of operations, the service requirements are met by the

Table 1.2: Characteristics of Sentinel-1 measurements mode [6].

Parameter	Interferometric Wide-swath mode (IW)	Wave mode (WV)	Strip Map mode (SM)	Extra Wide-swath mode (EW)
Polarisation	Dual (HH+HV, VV+VH)	Single (HH, VV)	Dual (HH+HV, VV+VH)	Dual (HH+HV, VV+VH)
Access (incidence angles)	31°–46°	23°–37° (mid incidence angle)	20°–47°	20°–47°
Azimuth resolution	<20 m	<5 m	<5 m	<40 m
Ground range resolution	<5m	<5 m	<5 m	<20 m
Azimuth and range looks	Single	Single	Single	
Swath	>250 km	Vignette 20×20 km	>80 km	>410 km
Maximum NESZ	−22 dB	−22 dB	−22 dB	−22 dB
Radiometric stability	0.5 dB (3σ)	0.5 dB (3σ)	0.5 dB (3σ)	0.5 dB (3σ)
Radiometric accuracy	1 dB (3σ)	1 dB (3σ)	1 dB (3σ)	1 dB (3σ)
Phase error	5°	5°	5°	5°

Table 1.3: Sentinel-1 payload parameters [6].

Parameter	Value
Centre frequency	5.405 GHz
Bandwidth	0 … 100 MHz (depending on the mode and sub swath)
Polarisation	HH+HV, VV+VH
Antenna size	12.3 m×0.821 m
RF Peak Power (sum of all TRMs, at TRM o/p)	4368 W
Pulse width	5–100 μs (programmable)
Transmit duty cycle:	
Max	12%
SM mode	8.5%
IW mode	9%
EW mode	5%
WV mode	0.8%
Receiver noise figure at TRM input	3.2 dB
Pulse repetition frequency	1000–3000 Hz (programmable)
ADC sampling frequency	300 MHz (real sampling) (Digital down-sampling after A/D conversion)
Sampling	10 bits
Data compression	Selectable according to FDBAQ
Instrument operation	Up to 25 min per orbit continuously in any of the imaging modes and for the rest of the orbit in WV mode
Instrument mass	945 kg
CSAR DC power	3870 W (IW mode, two polarisations)
CSAR data storage capacity	1410 Gb (End-of-Life)
Effective CSAR downlink data rate	Two channels of 260 Mbps each

nominal imaging mode, the Interferometric Wide-swath (IW) mode, in combination with the Wave (WV) mode. Also, the mutually exclusive Strip Map (SM) mode and Extra Wide-swath (EW) mode have been equipped for continuity reasons. Then, two mutually exclusive dual-polarization modes are provided for the imaging modes. Their main characteristics are shown in table 1.2.

The CSAR instrument carried as payload for the mission is a synthetic aperture radar operating in C-band, with horizontal and vertical polarization in transmission and reception. It is made up of two main components: the SAR Antenna Subsystem (SAS) and the SAR Antenna Subsystem (SAS) [6]. The key parameters are shown in the table 1.3.

The active phased antenna array provides fast scanning capabilities in both elevation and azimuth directions to respectively allow ScanSAR operations and

use of TOPSAR technique. In this way, it can cover a wide range of incidence angles and reach image quality requirements.

Then, the payload is equipped with dual-channel transmission and reception modules and pairs of slotted waveguides with H/V polarization. An internal calibration scheme allows monitoring of amplitude and phase variations for later processing adjustments, since transmit signals are routed into the receiver; this results in high radiometric stability. This capability is also backed by radiating waveguides[1] made of metalized carbon fiber reinforced plastic.

Taking into account that ground range resolution depends on the incidence angle, a digital chirp generator and selectable receive filter bandwidths permit efficient use of on-board storage. This aspect is further supported by a Flexible Dynamic Block Adaptive Quantization (FDBAQ), which also minimizes downlink times.

Downlink The Sentinel-1 Payload Data Ground Segment (PDGS) receives CSAR instrument data and provides systematic processing, archiving and dissemination of operational SAR data. It also ensures different planning and monitoring tasks such as CSAR instrument activities and X-band downlink operations. In this way, real-time and played-back data are downlinked to ground via the European Data Relay Satellite (EDRS), and, once received, they are down-converted, demodulated and transferred to the processing facilities. What follows is the systematic generation and archiving of higher-level data; during this phase, frame synchronization and reconstruction of CSAR data are performed through Instrument Source Packets (ISP) in a Computer Compatible Format (CCF) [6].

Data products Data are delivered as payload level-0. They are then processed to produce Level-1 and -2 products applying the required algorithms and formatting techniques. All products are broadcasted in the Sentinel Archive Format for Europe (SAFE) format. The different products are available in single or dual-polarization, according to the mode of operation; each mode can potentially generate products at different levels [7]:

- Level-0 SAR: it consists of SAR raw data, unfocused and compressed using FDBAQ sequence;
- Level-1: this type of data is generally used by most of the users; it differentiates into two main categories:

[1] A waveguide is a structure that guides waves, such as electromagnetic waves or sound, with minimal loss of energy by restricting the transmission of energy to one direction.

  - Single Look Complex (SLC) products are focused on SAR data geo-referenced with orbit and attitude data from the satellite and delivered in zero-Doppler slant-range geometry. It includes a single look in each dimension thanks to the full transmit signal bandwidth and is composed of complex samples preserving the phase information.
  - Ground Range Detected (GRD) products are focused on SAR data that has been detected, multi-looked and projected to ground range using an Earth ellipsoid model. In this case, phase information is lost, and the result has approximately square spatial resolution pixels and square pixel spacing with reduced speckle, at the cost of worse spatial resolution. The products are delivered in three resolutions: Full resolution, High resolution and Medium resolution.

- Level-2 Ocean (OCN): it consists of geolocated and geophysical products derived by the previous level; they contain components needed for wind, wave and currents applications: Ocean Wind field (OWI), Ocean Swell spectra (OSW) and Surface Radial Velocity (RVL).
  - OSW is a two-dimensional ocean surface swell spectrum and includes an estimate of the wind speed and direction per swell spectrum;
  - OWI is a ground range gridded estimate of the surface wind speed and direction at 10 m above the surface, and it is derived from Level-1 GRD images in SM, IW or EW modes;
  - RVL is a ground range gridded difference between the measured Level-2 Doppler grid and Level-1 calculated geometrical Doppler.

Availability Processed Sentinel-1 data are available globally, regionally and locally within a defined timescale through a dissemination network. Data are delivered within an hour from sensing for Near Real-Time (NRT) emergency response, three hours for NRT priority areas and 24 hours for all acquired data; NRT data requires a subscription. Global products are systematically generated for all required data; regional ones are systematically generated over well-defined regions. If data are required during critical surveillance cases or for national services, the satellite sends the required data, which are received by local ground stations involved in the service. Overall, the observation scenario is pre-defined, and the production is regular [7].

Table 1.4: Sentinel-2 mission features [6].

Mission features	Data
Mission lifetime	15 years
Number of satellites	2
Nominal in orbit satellite lifetime	7.25 years with consumables for additional 5 years
Nominal orbit	Sun synchronous 786 km (mean altitude), 10:30 LTDN
Land coverage	$-56°$ to $+84°$
Global revisit time	<5 days
Global NRT latency	<2 h to reception on ground
High quality mission products	Level 0, 1
Mission phases	LEOP, commissioning, operational, deorbiting

Sentinel-2

In the GMES environment, the Sentinel-2 mission provides continuity to services relying on multispectral high-resolution optical observations over global terrestrial surfaces. The objectives for Sentinel-2 are:

- To provide systematic global acquisitions of high-resolution multispectral imagery with a high revisit frequency;
- To provide enhanced continuity of multispectral imagery provided by the Satellite Pour l'Observation de la Terre (SPOT) series of satellites;
- To provide observations for the next generation of operational products such as land-cover maps, land change detection maps, and geophysical variables.

Sentinel-2 sustains the operational supply of data for services such as Risk Management (floods and forest fires, subsidence and landslides), European Land Use/Land Cover State and Changes, Forest Monitoring, Food Security/Early Warning Systems, Water Management and Soil Protection, Urban Mapping, Natural Hazards, and Terrestrial Mapping for Humanitarian Aid and Development [8].

The Sentinel-2 mission offers an unprecedented combination of systematic global coverage of land surfaces, high spatial resolution, and a wide field of view. The two satellites that compose the mission are in a Sun-synchronous at 786 km altitude, which guarantees five days revisit time at the equator. The two identical satellites operate simultaneously, favoring a small, cost-effective and low-risk solution. Features are shown in table 1.4 [8].

The combination of the large swath of 290 km, spectral range, coupled with the global and continuous acquisition requirement with high-revisit frequency, leads

Table 1.5: Sentinel-2 system parameters [6].

System features	Data
Evolution	Need for 4 satellites to fulfill 15 years of operations
Ground station scenario (for payload data recovery)	Kiruna, Svalbard, Maspalomas, Prince Albert, plus Local user stations
Security	Authentication of commands
Reliability	>0.7
Availability	>97%
Geo-location (2sigma)wo GCP	<20 m
Swath	290 km
Modes of operation	Nominal mode, support modes, safe mode
Maximum imaging	32 min per 100 min orbit
Nominal launcher	Rockot (1225 kg)
Back-up launcher	VEGA (1226 kg)
Satellite launch mass	1225 kg (with 70 kg margin included)
Satellite dimensions (stowed)	3390 mm×1630 mm×2350 mm

to the daily generation of about 1.6 TBytes of compressed raw image data from the constellation. The primary data related to the Sentinel-2 system definition is presented in table 1.5 [8].

The Sentinel-2 satellite is based on a relatively compact design that ensures compatibility with small launchers like VEGA and Rockot. The satellite lifetime is specified as 7.25 years, including a 3-months commissioning phase, and the propellant is sized for 12 years, including provision for deorbiting maneuvers at end-of-life. The satellite is three-axis stabilized with an Attitude and Orbit Control System (AOCS) based on an advanced multi-head star tracker, a laser gyroscope and a dual-frequency Global Navigation Satellite System (GNSS) receiver. The top panel is providing accommodation for the MultiSpectral Instrument (MSI). The critical sensor assembly carrying the three-star trackers and the laser gyroscopes package are accommodated on top of the payload instrument [8].

The MSI is the main feature of Sentinel-2 satellites, and 13 spectral bands span from the Visible (VIS) and the Near-Infrared (NIR) to the Short Wave Infrared (SWIR) at different spatial resolutions at the ground ranging from 10 to 60 m (bands defined in table 1.6). [8]. The system works passively by collecting sunlight reflected from the Earth. New data is acquired at the instrument as the satellite moves along its orbital path. The MSI uses a push-broom concept.[2] The average period of observation over land and coastal areas is approximately 17 minutes, and the maximum period of observation is 32 minutes. Light reflected

[2]A push-broom sensor works by collecting rows of image data across the orbital swath and utilises the forward motion of the spacecraft along the path of the orbit to provide new rows for acquisition [9].

Table 1.6: Sentinel-2 spectral bands definition [6].

Band number	Central wavelength (nm)	Band width (nm)	Lref (Wm^{-2} sr^{-1} μm^{-1})	SNR @ Lref
1	443	20	129	129
2	490	65	128	154
3	560	35	128	168
4	665	30	108	142
5	705	15	74.5	117
6	740	15	68	89
7	783	20	67	105
8	842	115	103	174
8b	865	20	52.5	72
9	945	20	9	114
10	1380	30	6	50
11	1610	90	4	100
12	2190	180	1.5	100

up to the MSI instrument from the Earth and its atmosphere is collected by a three-mirror telescope and focused, via a beam-splitter, onto two Focal Plane Assemblies (FPAs): one for the ten NIR wavelengths and one for the three SWIR wavelengths. Radiometric calibration of the MSI instrument is achieved via a diffuser fitted on the inside face of the combined Calibration and Shutter Mechanism (CSM) [9].

Downlink The PDGS is responsible for payload and downlink planning, data acquisition, processing, archiving and downstream distribution of the Sentinel-2 satellite data [8].

Data products Sentinel data products are made available systematically and free of charge to all data users, including the general public, scientific and commercial users [10].

They are as follow [8]:

- Level-0 and -1A: they consist of raw compressed and uncompressed data, respectively;
- Level-1B: this data is radiometrically corrected, and its physical geometric model is refined;
- Level-1C: this product provides geo-coded Top Of Atmosphere (TOA) reflectance with a sub-pixel multispectral and multi-date registration, and associated cloud and land/water masks;
- Level-2A: through a software toolbox on the user side, Bottom Of Atmosphere (BOA) reflectance is derived, and cloud masks are enhanced.

1.3.2 The in-situ component

Copernicus also relies on in-situ data that is on-site or from local measurements, mostly belonging to the EU Member States or international research infrastructures, and which are made available to Copernicus through agreements.

It includes observations from the ground, sea and airborne sensors, as well as geospatial references and ancillary data licensed or provided for use in the Copernicus program. More recently, in-situ data has found new sources in sensors and imagery gathered by Unmanned Aerial Vehicles (UAVs) or crowdsourcing (e.g., OpenStreetMap).

In-situ data has two roles:

- To calibrate, supplement and validate satellite mission data to ensure it is delivering sustainable and reliable data over time;
- To help Copernicus service operators and the space component to produce products and deliver services that are requested by end-users.

When multiple services require the same data, the data access is coordinated by each Copernicus service operator or by the EEA [3].

1.4 Services

Copernicus possesses six services that address different thematic areas, corresponding to the daily needs of European citizens: Atmosphere, Marine Environment, Land, Climate Change, Emergency Management and Security. The services exploit Sentinel data, contributing mission data or in-situ data to provide products and information that support commercial, institutional and research applications [3].

Copernicus Atmosphere Monitoring Service CAMS is managed by the ECMWF and has the aim to continuously monitor the composition of the Earth's atmosphere at global and regional scales through the provision of near-real-time data and forecasts products. It is mostly used for health, renewable energy, or climatology issues.

Copernicus Marine Environment Monitoring Service CMEMS is managed by *Mercator Océan International*, to provide regular and systematic information about the physical state and dynamics of the ocean and marine ecosystems.

Its products cover the global oceans and the European regional seas, through the provision of observations and forecasts. It is mostly used for ship routing services, offshore operations or aquaculture.

Copernicus Land Monitoring Service CLMS has the aim to provide geographical information on land cover, land use, land cover-use changes over the years, vegetation state and the water cycle. It is mostly used for forest management, water management, agriculture or food security.

Copernicus Climate Change Service C3S is managed by the ECMWF. It has the aim to respond to changes in the environment and society associated with climate change, through the provision of information for monitoring and predicting climate change and help to support adaptation and mitigation strategies. It is mostly used for climate, weather and renewable energy monitoring.

Copernicus Emergency Management Service CMES encompasses two components: the early warning component is managed by the Directorate-General (DG) JRC, and the mapping component is managed by DG for European Civil Protection and Humanitarian Action (ECHO). The aim of the former is to deliver warnings and risk assessments of floods and forest fires, while the mapping service provides map and geo-information products for all types of natural and human-made disasters. CMES intervenes both at European and global levels.

Copernicus Security Service The Copernicus Security service aims at improving crisis prevention, preparedness and response in three domains: border surveillance (managed by FRONTEX), maritime surveillance (managed by EMSA) and support to EU External Action (managed by EU SatCen). It is mostly used to support related European Union policies by providing information in response to the security challenges Europe is facing.

1.5 Data availability

Sentinel data is provided on an open and free basis by ESA and the European Commission to promote full utilization, increased scientific research, growth in EO markets and job creation. This system is based on the following principles [9]:

- Anyone can access acquired Sentinel data. In particular, no distinction is made between public, commercial and scientific uses, or between European or non-European users.
- Licenses for the use of Sentinel data are available free of charge.
- Sentinel data will be made available to users via generic online access, free of charge, subject to a user registration process and acceptance of generic terms and conditions.
- Additional access modes and delivery of additional products will be tailored to specific user needs and will be subject to tailored conditions.
- In the event that security restrictions apply, affecting the availability or timeliness of Sentinel data, specific operational procedures will be activated.

Large scale online distribution of the Sentinel data is possible within the technical limitations of the system (in particular the dissemination network) and depends on operational funding levels.

Open and free access to the data maximizes the beneficial utilization of Sentinel data for the widest range of applications.

Chapter 2

Software background

> *May not machines carry out something which ought to be described as thinking but which is very different from what a man does?*
>
> **— Alan Turing**

The main idea of the project is taken from [11]: it moves from raw data, i.e. satellite images, and goes through a neural network model that, employing deep learning techniques, has learned how to simulate the optical image.

Raw data, both optical and SAR, come in large images that carry with them all geometrical and spectral errors. So they need to be processed to remove them and then co-registered to make images of different bands with the same size and related to the same area: each pixel in every image has to be in the same spot and match with its corresponding pixel in the other bands. The processed images are the source of the dataset that must be created to allow the code to work.

Following the pre-processing of data, the neural network has to be created, trained, tested, and then saved. Few words that in terms of coding and complexity mean a lot because the model has to *learn* how to simulate a complex pattern, i.e. the original image to be replicated.

The software Sentinel Application Platform (SNAP) has been used for the pre-processing, while the neural network and the side scripts have been coded in Python, mainly using the PyTorch library. The theory behind the project is covered here, while the actual workflow is described in the next chapter.

2.1 Sentinel Application Platform

The Sentinel Application Platform (SNAP) is a common architecture for all the toolboxes related to sentinel imagery and it is provided by the European Space Agency (ESA). The main purpose is Earth Observation processing and analysis in the field of remote sensing space missions. In this way, a unique environment is used to work on different types of images from different missions, not only Sentinel [12]. It is a free software licensed under the GNU GPL.

The toolboxes are different, but in this project only those for Sentinel-1 and -2 are used.

The Sentinel-1 Toolbox (S1TBX) consists of a collection of processing tools, data product readers and writers and a display and analysis application to support the large archive of data from ESA SAR missions including SENTINEL-1, ERS-1 and -2 and ENVISAT, as well as third party SAR data from ALOS PALSAR, TerraSAR-X, COSMO-SkyMed and RADARSAT-2 [13]. It also includes tools for calibration, speckle filtering, co-registration, orthorectification, mosaicking, data conversion, polarimetry and interferometry.

The Sentinel-2 Toolbox (S2TBX) comprises a rich set of visualization, analysis and processing tools for exploiting optical high-resolution products. It is a multi-mission remote sensing toolbox, so provides support for data from third party missions [14].

2.1.1 Processing explanation

The software SNAP was used during the pre-processing phase, where the data needed to be refined before being processed by the algorithm. The flow was the same as [11]: calibration, despeckling and Range Doppler Terrain for SAR images; then, co-registration of optical and processed radar images. This section details their background, while the next chapter contains the details regarding the project.

Calibration

SAR data needs to be calibrated to allow further processing and work. Radiometric calibration corrects a SAR image so that the pixel values truly represents the radar backscatter of the reflecting surface [15]. The information required to apply the calibration equation is included within the Sentinel-1 GRD product; specifically, a calibration vector included as an annotation in the product allows simple conversion of image intensity values into sigma-naught values. The calibration

reverses the scaling factor applied during level-1 product generation, and applies a constant offset and a range-dependent gain, including the absolute calibration constant [16].

Speckle filtering

Speckle, appearing in SAR images as granular noise, is due to the interference of waves reflected from many elementary scatterers. Speckle in SAR images complicates the image interpretation problem by reducing the effectiveness of image segmentation and classification [17].

When such a procedure is done at an early processing stage of SAR data, speckle is not propagated in ongoing processes (i.e., terrain correction) [16]. The refined Lee filter has been found to be superior, with respect to other single product speckle filters, for visual interpretation, because of its ability to preserve edges, linear features, and point target and texture information [17]. More recently, multitemporal speckle filters have been developed to reduce speckle, taking advantages from multiple SAR observations in time [16].

Range Doppler Terrain

SAR data are generally sensed with a varying viewing angle greater than 0 degrees, resulting in images with some distortion related to side-looking geometry (i.e., layover, shadows and foreshortening). Terrain corrections are intended to compensate for these distortions so that the geometric representation of the image will be as close as possible to the real world. Range Doppler Terrain correction is a correction of inherent geometric distortions caused by topography, such as foreshortening and shadows, using a Digital Elevation Model (DEM) to correct the location of each pixel [16]. The DEM geocodes the image, which is converted from slant range into a map coordinate system.

The Range Doppler Terrain correction available in SNAP makes use of available orbit state vector information in the metadata, the radar timing annotations, and the slant to ground range conversion parameters together with the reference digital elevation model data to derive the precise geolocation information [16].

The system permits choosing the Coordinate Reference System (CRS), the image resampling method and the pixel spacing for the target product; thus, it allows the spatial snapping of Sentinel-1 Ground Range Detected (GRD) products to Sentinel-2 MultiSpectral Instrument (MSI) data grids, in order to geolocate data to a common spatial grid and promote the use of satellite virtual constellations [16].

Co-registration

The process of co-registration ensures that each ground target contributes to the same pixel in all the images in terms of range and azimuth. The single images differ in time and angle and are not related to the same path exactly since different satellites usually have different orbits. All images must have the same size and pixelation and must all relate to the same path of terrain; thus allowing them to be stacked together.

To merge data from two different satellites, co-registration through reprojection was used. This process was possible with the help of metadata (i.e., orbital data, reference DEM, CRS) linked to the images or available automatically in the settings; in this way the output images had the same size and were related to the same CRS, thus they could be stacked together and enhance the information available.

2.2 Deep learning

The importance of data in all their forms has taken off during the recent years, and it seems to increase and become more pervasive as technology advances: it is possible to think about new words as big data, or rather new subjects as data science, and how they are shaping the world of tomorrow. The ideas related to artificial intelligence have also advanced in order to allow the processing and the usage of more extensive databases.

Artificial intelligence is defined as "the theory and development of computer systems able to perform tasks normally requiring human intelligence" [18]. It also includes any method used by a machine to emulate human behavior [19]. In this way, AI uses machine learning techniques that allow "the computer to learn without being programmed to do so" [19]. In this framework, it is possible to define deep learning as a machine learning technique that consists of algorithms, which are also called neural networks, "that permit software to train itself" on a large amount of data and understand the patterns and features inside them [20]. We can then use the trained algorithm, also called the model, to make predictions or inform future decisions [19].

Even though linking the idea of *learning* and *understanding* to a computer is not trivial, the machine actually learns and adapts itself to the data proposed. It is possible to imagine the algorithm as a black box that takes some inputs and generates one or more outputs according to the task. This process is what actually happens in our brain where we use past experiences, or rather past outputs, to

train and so modify our present and future behavior to make it closer to our expectations.

2.2.1 Artificial neural networks

Perceptron

A straightforward path to start understanding neural networks moves from the idea of a single neuron, also called a perceptron. From here, it is possible to create even more extensive and complex algorithms by adding more neurons or blocks to the network or increasing its inner complexity by using different operations.

A neuron is the fundamental building block of complex algorithms. It consists of a set of operations applied to one or more inputs to release an output: this process is called feedforward propagation, and it is shown in figure 2.1. At first, a linear combination of the inputs is made, so every input x is multiplied for its associated weight w, and the results are summed together with a bias term b; the bias is used to shift the following function since b is not related to any input. By applying linear algebra to the problem, it is also possible to assort inputs and weights into vectors and write the linear combination as a dot product between them. Finally, the result of the summation is passed through an activation function that generates the output $\hat{y}$.

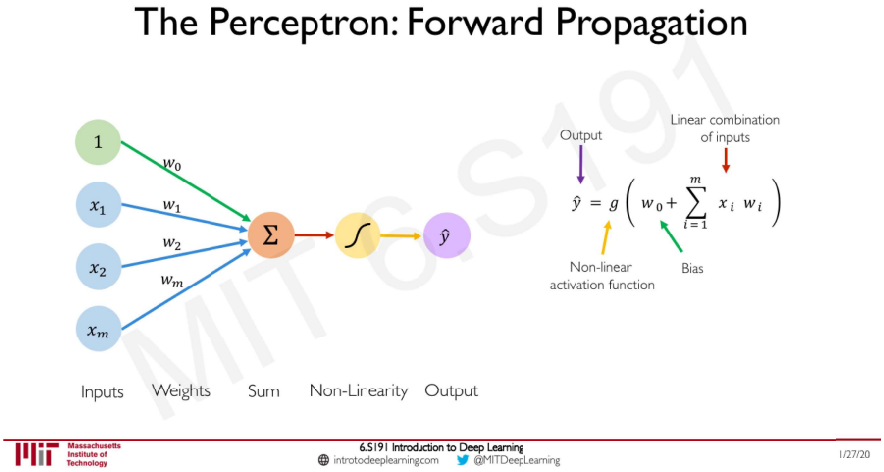


Figure 2.1: Perceptron visualization with linear combination [19].

The aim of the activation function is twofold:

- Shape the output as a known function to it is easier to predict: for instance, limiting the output in the range $[0 : 1]$ to represent a probability;
- Add non-linearity through the usage of a non-linear activation function that helps to shape the model to understand non-linear patterns in the data.

Deep neural networks

The next step is adding a new neuron, connected to all the inputs through its weights, and that performs the feedforward process as its counterpart, releasing its output. The layer made with these neurons is called dense since all outputs are connected to all the inputs. This model is called a multi-output perceptron.

Each group made with similar items is called a layer, so an input layer contains all the inputs, an output layer all the outputs, and a hidden layer all the neurons in the same position.

Therefore, a neural network is a network made of one or more hidden layers densely connected with each other and with the input and output layers (figure 2.2). Each group of connections is a transformation of the previous values into the new ones, so it has its own set of weights.

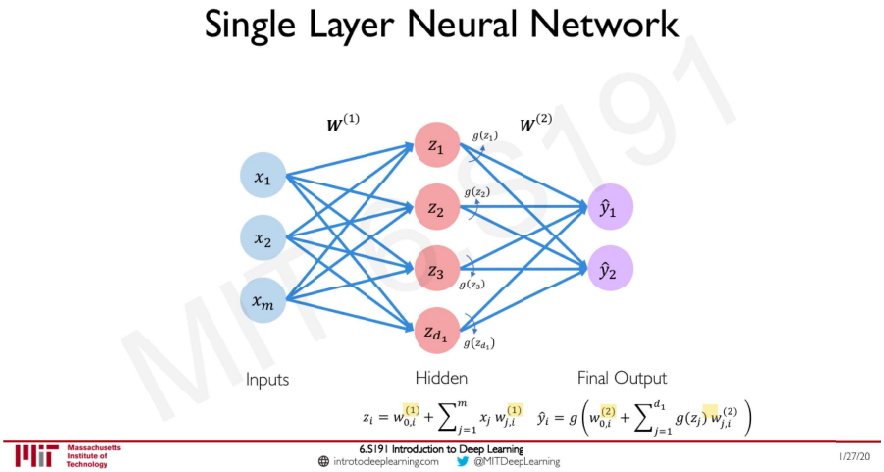


Figure 2.2: Neural network with a single hidden layer [19].

Following this process of enlarging the network, a deep neural network is a network where we added more fully-connected, i.e. dense, hidden layers.

Convolutional Neural Network (CNN)

Increasing the complexity, together with the depth of a neural network, allows the algorithm to perform more tasks at a higher level. This happens when we start dealing with computer vision, since images are usually big in size and cannot be represented as vectors because they are matrices of pixels. Then, there is the need to maintain spatial information, since it is linked to the whole meaning of the image: the change in pixel values will show different things and usually the localized features that create what is shown are made of groups of pixels that together make sense, so the algorithm must be adaptable in recognizing what is depicted. In this way, a straightforward task for computer vision is classification, which is the capability to categorize the input images assigning the correct label; this task is part of the project too [19].

For a computer, an image is a matrix of pixels; in the case of colored images, it is a set of three matrices stacked together, one per each color channel (Red, Green and Blue). So the input is two-dimensional.

This stage is where a convolutional neural network takes over. A convolutional layer replaces the hidden layer and performs convolution instead of a linear combination. It is adaptable and can be used on multidimensional inputs too.

The process starts and ends with a two-dimensional matrix. The convolutional layer has a set of filters, also called kernels, that are little matrices of given size filled with given values; these filters are the weights of the layer, and their values represent the feature they are capable of detecting. The operation consists in overlaying the kernel on a patch of the input image, performing element-wise multiplication between the values in the kernel and the related values of the input, summing up all the element-wise products and repeat the process shifting the kernel of a given stride until the image is covered entirely. The results of the summations, shown in figure 2.3, create the output matrix preserving the spatial information since the position of a value in the output matrix is related to the position of the kernel on the image during the operation [21].

The previous operations apply for one kernel, but more than one is needed since the model has to detect more than one feature. In this way, the convolutional layer can be represented as a volume that has three dimensions: two for the width and height, and one for the depth of the layer. The depth is the number of filters

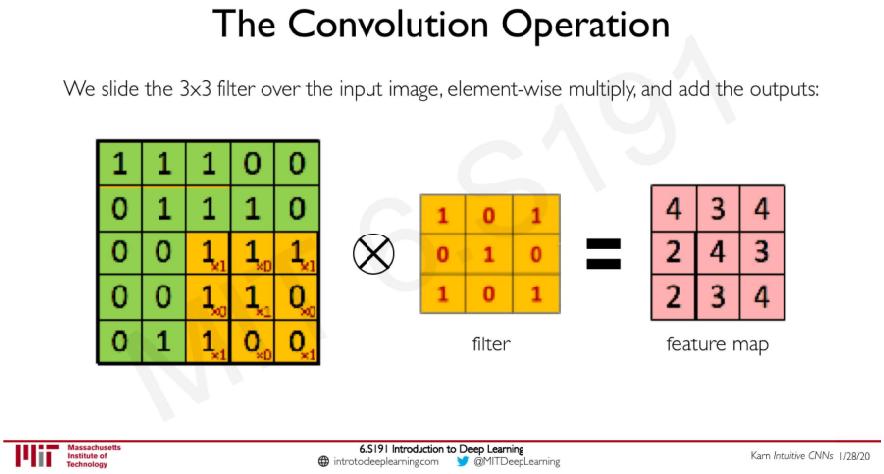


Figure 2.3: Result of a simple convolution operation on images [19].

applied in that convolutional layer.

However, convolutional neural networks do not have only convolutional layers. They still need activation functions to deal with non-linearity, and a pooling layer is usually used to downsample the image, but this is not used in the project. According to [22], [23], Batch Normalization (BN) and dropout layers have been added after the convolutional layer.

According to [22], BN helps in increasing the accuracy of the model while making the training faster. It is a transform that normalizes the output splitting it into batches and using the batch mean and variance; then, it scales and shifts the normalized value by two parameters. These last two parameters are learnable, so they are updated through training. The algorithm is in figure 2.4.

Regarding the *dropout* layer, as for [23], it is used as a regularization[1] method to improve the performance of deep neural networks. It consists in shutting down randomly some units with their connections during training to prevent overfitting.[2] The amount of units to drop is set as a probability of the dropout layer to retain a hidden unit [22]. An example is shown in figure 2.5.

[1]Regularization is related to methods to improve the training of the model, and so helping the model in learning the data instead of just memorizing them. This way improves the generalization of the model [19], [24].

[2]The process through which the model adapts and so understands the data is known as fitting. Overfitting is a potential issue during training where the model has adapted too much

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

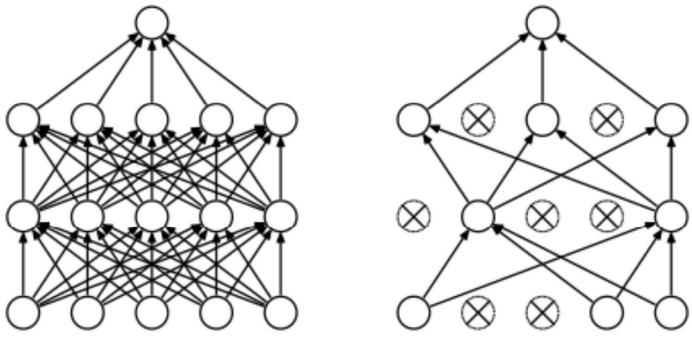
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Figure 2.4: Algorithm for Batch Normalization transform [22].



(a) Standard Neural Net (b) After applying dropout.

Figure 2.5: Example of the application of dropout on a network [23].

Residual Network (ResNet)

A residual network is a network architecture where a shortcut connection is created from the input and the output of the main block of the original network, just before passing the output into the activation function [25]. ResNets have been added in the project model to replicate the idea of [11], according to the papers [26], [27]. This type of network has been proved to reduce training errors in very deep neural networks due to the difficulty of the network to converge to the ideal value.

In the project model, the input is added to the initial output of a convolutional block, and then the result is passed through the activation function. This structure is also called ResNet block in the project, where many blocks of this kind are stacked together. The addition is possible since the input and the initial output have the same shape.

to the data that its accuracy on new predictions decreases; or rather its ability to generalize and manage unseen data decreases [19].

The idea behind the increased performance is based on the residual learning framework, shown in figure 2.6. The function $f(x)$ encloses the operations made in the main block, where x is the input added through the connection, and $H(x) = f(x) + x$ is the original function that the layers have to fit during training, i.e. the output of the entire ResNet block. Then, the stacked weight layers that make up the main block are forced to learn the residual function $f(x) = H(x) - x$, that is what is updated through the training process [25].

In this project, ResNeXt architecture is used instead: this is an evolution that modifies how the ResNet framework is applied, not the results of it. The modified learning block is in figure 2.7. As for the paper [28], computational complexity and model size are the same. The difference is that the operations performed in the main block are split in several paths of lower dimension; the results are regrouped before adding the input through the shortcut. The number of paths is expressed with a new parameter called cardinality. This idea is developed in the project by using grouped convolutions. Overall, it increases accuracy against ResNets and allows for parallel computing on different processing units [28].

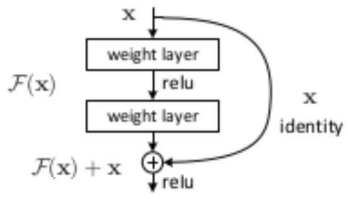


Figure 2.6: Learning block for the residual framework [25].

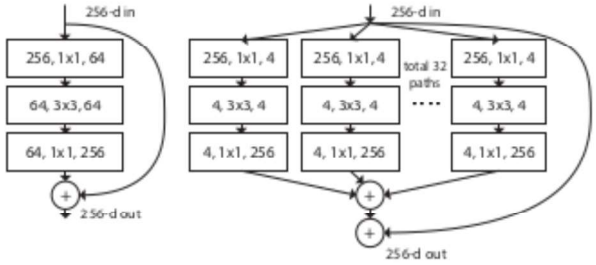


Figure 2.7: Difference between the ResNet framework and its ResNeXt counterpart with cardinality=32 [28].

Activation functions

As said before, an activation function is used to make the output of a neuron more predictable and insert non-linearity in the model to fit the data better. It can also be applied to multidimensional data in complex networks. Following are some

non-linear functions applied in the project. All these functions are differentiable since the gradient is needed for training.

Sigmoid or Logistic The main reason this function is used is that the output range is $[0 : 1]$, so it can represent a probability. The Sigmoid function is particularly useful in classification problems where the model has to assign a probability as output. It is applied at the end of the project model and together with the loss function.

Tanh The hyperbolic tangent belongs to sigmoidal functions, but the output range is $[-1 : 1]$. It cannot represent a probability, but performs better in training [29] and is used as the last layer in a part of the project network as for reference [11], [30].

ReLu The simple Rectified Linear (ReLu) function is both computationally efficient and non-linear since it is not all rectified. It presents the Dying ReLu problem, which means that negative values are not mapped, and the model cannot learn from them.

The Leaky Rectified Linear (LReLu) function is used to address the issue. It adds a little slope for negative values increasing the range of ReLu to negative infinity and helping in training [31]. These two functions are used in the project, as for reference [30].

2.2.2 Machine learning

The machine learning process is defined by [32]: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." The variety of tasks is what makes these algorithms so powerful and attractive.

Machine learning algorithms can also be categorized as supervised and unsupervised according to the experience:

- Supervised learning algorithms learn from a dataset where each example is associated with a label, so the structure of the data is given, and the model tries to replicate the function mapping examples to labels;
- Unsupervised learning algorithms learn the features and properties of the underlying structure of the data where examples do not have labels, so learn

the features from the dataset itself.

Training

When a neural network is created, its parameters are initialized randomly or following some methods, but it has not adapted to data yet. The algorithm is not able to make the right prediction at this point since it has not learned how to replicate the data behavior and their inner patterns.

The algorithm has to evaluate how much its current prediction, i.e. end output, is distant from the dataset value it is trying to emulate, that is called ground truth; this distance is calculated through the cost, or loss, function, and is a direct measure of the performance of the network. The function is generally applied to all samples, so the whole cost function is the average on all samples of the single loss functions. It also typically includes at least one term to perform statistical estimation and may include regularization terms [24]. Then, the neural network needs a way to update all its weights according to the loss: many algorithms exist for the so-called optimization procedure, and they are usually related to the task of the network.

Regarding the loss function, two main types are used in the project, as for [11]: Binary Cross Entropy (BCE) loss and L1 loss. They are discussed here, while their implementation is in the next chapter.

The **BCE loss** (equation 2.1) is usually used when the model output a probability between 0 and 1. This function is used in [11] and is adapted to the objective of the project.

$$l(x, y) = L = \{l_1, \dots, l_N\}^T, \quad l_n = -w_n[y_n \cdot \log(x_n) + (1 - y_n) \cdot \log(1 - x_n)] \quad (2.1)$$

The **L1 distance** (equation 2.2) is borrowed from math to produce a measure of distance between two tensors in a multidimensional space; that is what happens in the project where RGB images have to be compared. The loss in the code measures the mean absolute error between each element in the input and target [33].

$$l(x, y) = L = \{l_1, \dots, l_N\}^T, \quad l_n = |x_n - y_n| \quad (2.2)$$

Since the loss function shows how far the output is from the ground truth, the training process aims to update the parameter in order to find those that minimize the loss: this is an optimization problem, solved in general with the so-called gradient descent. This method drives the cost function to very low values in order to minimize it so that the output is very similar to the reality [24]. The

algorithm is shown in figure 2.8.

The process starts from the cost function that is the average of the loss calculated on the prediction, based on inputs and weights, and the ground truth. This function is differentiable and has some minima. In order to train the network, the gradient of the cost function is calculated, and the weights of the network are updated using the opposite of the gradient times a parameter called the learning rate. The opposite is used to converge to a minimum, while the learning rate is used to scale the updating term and define how significant is the updating step at each iteration. The objective is to reach the absolute minimum of the function while avoiding local minima.

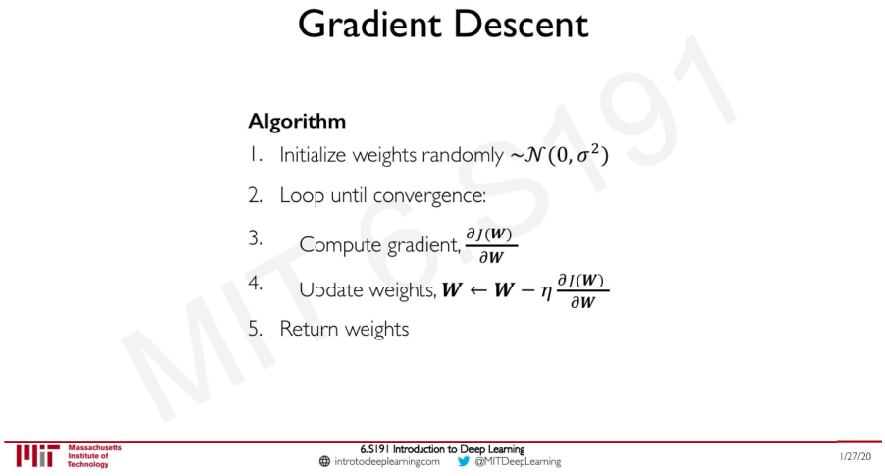


Figure 2.8: Fundamental algorithm to accomplish gradient descent [19].

Since the gradient of the cost function is related to the final output and the ground truth, it is possible to calculate it through backpropagation: this is an iterative process that evaluates the gradient at each step of the network using the chain rule applied to the partial derivative of the loss [19], [21], [24].

Many algorithms exist for optimization and can be used to complete and integrate the learning process. In the project, the Adam optimizer is used [11]. This gradient-descent algorithm proved to be "straightforward to implement, computationally efficient and well suited for large problems in terms of data or parameters" [34].

The training process is generally followed by a validation phase, where different models trained on the same dataset are evaluated to find the one that is the best, and the test phase, where the model is tested to get performance data [35]. The project model has been taken from [11], so the hyperparameters[3] are set the same.

The test phase had been replicated using the large source image split into two parts, so the test and training datasets are different but have the same distribution. Two evaluation indicators are used to assess the quality of the generated image: Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity (SSIM) [11].

2.2.3 Deep generative model

A generative model is an algorithm that can generate new data; the new samples are linked to the input data. The input data has some probability distribution, so the model is trained to approximate that distribution; in this way, the algorithm generates an output with a distribution close to the input samples one. This type of model is essential because it can uncover underlying features and improve training by generating outliers in the distribution [19].

The model of this category used in this project is a Generative Adversarial Network (GAN). This framework is proved to work effectively, as for [11], [19], [24], [36]. A GAN is a large model made of two neural networks playing against each other:

- The generator network takes as input a noise variable and tries to replicate the distribution of training data by producing an output close to them;
- The discriminator network tries to distinguish between real samples from the training data and fake samples from the generator.

The output of the generator is like a sample of training data in shape, while the discriminator emits a probability value that indicates the probability that a sample is real or fake. Then, the discriminator is trained to classify samples as real or fake correctly. On the other hand, the generator wants to make an output closer to the real sample to convince the discriminator that the fake sample is real. This adversarial learning process is a zero-sum game where the two players aim to maximize their own payoff [24], [36]. The solution to this game is the Nash equilibrium.

[3]Hyperparameters are those outside the learning algorithm and are set using a validation dataset [24].

The functions and parameters involved in the process are as follow [36]:

- $P_z(z)$ is the distribution made of the input noise variables to feed the generator;
- p_g and p_d are the distribution of each network, the generator and discriminator respectively;
- G and D are two differentiable functions that represent the generator and discriminator network respectively;
- x is a sample of the data distribution p_{data}.

As said before, the two networks play a two-player minimax game over the function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log(D(x))] + \mathbb{E}_{z \sim z_{data}(z)}[\log(1 - D(G(z)))] \quad (2.3)$$

It is crucial to notice that $D(x)$ is the probability that x is a real sample from the data, while $D(G(z))$ is the probability that $G(z)$ is fake. In this way, D is trained to maximize the probability of assigning the correct label to both real training samples and fake samples from G. This happens maximizing the first term and minimizing the second one. Instead, G is trained to minimizing the second term by maximizing the value of $D(G(z))$; this happens if the discriminator is fooled to assign a real label to a fake sample. At convergence, the equilibrium is reached when D outputs 0.5 everywhere, that is when it is not able to distinguish real from fake samples.

This general explanation is adapted to the project where the samples are images, so 2D matrices of pixel values with a channel per color if colored.

The actual model of the project is an evolution of the basic GAN and is called Conditional Generative Adversarial Network (cGAN). It is from [11] and has been chosen because it can adapt well to the task of data fusion with multitemporal values. In this new framework, the conditioning is performed by feeding additional information into the networks [37]. Regarding this thesis project, a set of real values backs up the input noise when fed to the generator, so it maps a new function ($G(z, x) \rightarrow y_{fake}$ instead of $G(x) \rightarrow y_{fake}$); all values are multidimensional[4] images. In this way, x is a set of images related to the ground truth y that the generator is trying to replicate, and the input noise is backed up with specific values that find a reason in their link with the real image. This phase

[4]Three channels for RGB images, one for greyscale images.

is where the data fusion happens: the backup set is made of images of the same spot made at different times and using different wavelengths or spectra.

2.2.4 Project pipeline

Machine learning is more than just a bunch of parameters, layers and calculations. It is more comprehensive and creates a cyclical pipeline where data is the pivotal element (figure 2.9): so the entire project can be split into different parts that can be easily understood by following the flow of data. From this point of view, neural networks and training are just algorithms, blocks among others that build the pipeline. Every project has its own pipeline, and here this thesis project is used as an example. The background is from [38].

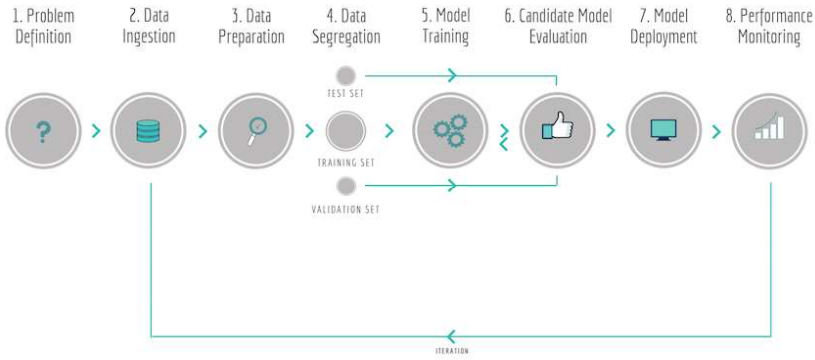


Figure 2.9: Visualization of a machine learning pipeline [38].

Everything starts with the definition of the problem, which contains the objective we want to achieve using machine learning techniques: this answers the question "What I need to do and how I can achieve it?". The problem was to fuse data from satellites using recent technologies and everyday resources aiming to reduce cloud covering in images or augmenting data. The purpose was to merely investigate the possibility to adopt machine learning in the field of dual-use[5] satellite images, exploring possibilities, further goals and resources needed to do that. There was also an analysis of the problem to find what model to implement, what data to use, how to manage them and what output expect.

First of all, data has to be collected and stored. The thesis model needed multi-temporal and both optical and radar images from Sentinel-1 and -2. Raw images were downloaded from the sentinel hub since they are free to use. Three places in Italy were chosen so that radar and optical images could be as close as possible in acquisition time and referenced to the same terrain. They are: Verona

[5]Assets used by both military and civilians.

and Lake Garda in the North; Rome and Lazio Coast in the Center; the middle of Sicily in the South. Two acquisition times per area were taken, and one optical (RGB) and two SAR (VH and VV) images per time. They were stored in the local drive.

Secondly, they needed to be prepared to be used. Using SNAP, SAR images were processed through calibration, despeckling, and range doppler terrain; then, radar and optical images were co-registered by reprojection [11]. This entire process allows us to reduce the errors in the radar images and creates definite-size image sets regarding the same path of terrain, i.e. the pixels in all the images of the same set relates to the same spot.

The next step is segregating data, that is splitting the data to create the training, validation, and test datasets that are fed to the model. The validation dataset was not used since hyperparameters were the same as [11]. Training and test images were from two different parts of the source images, so they could be similar but with a different distribution. A folder for every path of terrain was created with the images cropped from the larger source images. In this way, the datasets are subsets of the main image, which works as source data.

At this point, the model is trained using the training dataset. Here the algorithm is updated to recognize the features of the images and to replicate them. Usually, different models with different hyperparameters are trained on the same dataset, and the best one is validated on the validation dataset. Finally, the performance of the candidate model is tested using the test subset.

With the performance assessment, the thesis project ends. However, in a real-world scenario, the chosen model is "deployed and embedded in a decision-making framework." The model becomes a service of a broader environment, and the pipeline can be automated too. At this point, the model can start making new predictions or outputting new values starting from new inputs. Even though it is operational, its performance and accuracy are continually assessed and used to calibrate the model and inform further developments.

Chapter 3

Application

> *War is ninety percent information.*
>
> **— Napoleone Bonaparte**

3.1 Problem formulation

This thesis work aims to investigate the implementation of deep learning algorithms to accomplish a data fusion task on Sentinel imagery. The objective is to simulate an optical image using multi-temporal SAR-optical images. In this way, the simulation starts from different types of data taken at different timeframes: a SAR image is combined with a pair of optical and SAR images from a previous time to get an optical image. The resulting image is compared with an objective image, which is the optical one paired with the first SAR image.

A cGAN architecture is adopted to accomplish the task, which is also referred to as multi-temporal fusion based image simulation: the idea moves from [11] where this algorithm shows the best results.

The details of the work and the results are presented in the subsequent sections, and further development ideas are discussed at the end.

3.2 Source images description

Sentinel -1 and -2 raw data were downloaded from the Copernicus open access hub (`https://scihub.copernicus.eu/dhus/`). Raw data contain acquired SAR and optical images and metadata for all the bands of the sensor. Three areas in Italy were chosen and, for this thesis purpose, are referred to as North, Center,

Table 3.1: Acquisition times of source images.

	S0	O0	S1	O1
North	16 July 2018	18 July 2018	08 September 2018	11 September 2018
Centre	17 August 2018	19 August 2018	23 August 2018	24 August 2018
South	30 August 2018	31 August 2018	05 September 2018	05 September 2018

and South. Two acquisition dates per type of image were selected, creating two pairs of SAR-optical images per area (table 3.1).

At this stage, data were pre-processed using the SNAP software in the desktop version for Windows. SAR data consist of two bands (VV and VH polarisations) per date with a pixel spacing of 10 m; they followed the flowchart of calibration, despeckling, and Range Doppler Terrain. For optical images, three bands were chosen from Sentinel-2 raw data to create the RGB layers which compose a colored image (R - 665 nm, G - 560 nm, B - 490 nm) with a ground sampling distance of 10 m. SAR and optical images were co-registered by reprojection and saved in dedicated folders. Finally, the images were cropped as needed to remove parts without data: the final sizes for North, Center and South are 10980x9280, 10980x10980 and 10980x10240 respectively. The optical images to simulate are presented in figure 3.1.

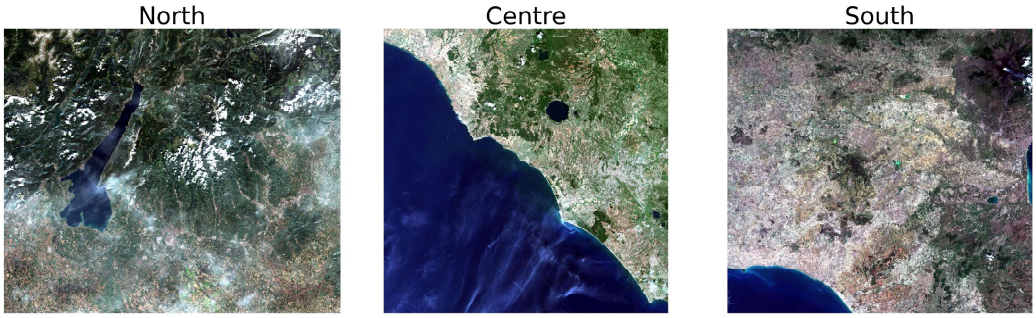


Figure 3.1: Optical images used as objective for the network.

The actual dataset to feed to the network is created by cropping patch sets of size 128x128 randomly from the source images; this dimension was chosen to allow a faster training according to the machine used. Each patch set consists of four patches from the same area of the source images; the second optical image is the ground truth, while the rest becomes the input of the algorithm. The training dataset is taken from the red area, while the test data from the yellow one; the test area is 25% of the source width (figure 3.2). The names set for the images are O0 and O1 for optical ones and S0VV S0VH S1VV S1VH for SAR ones.

The datasets are loaded before feeding them to the network using a custom

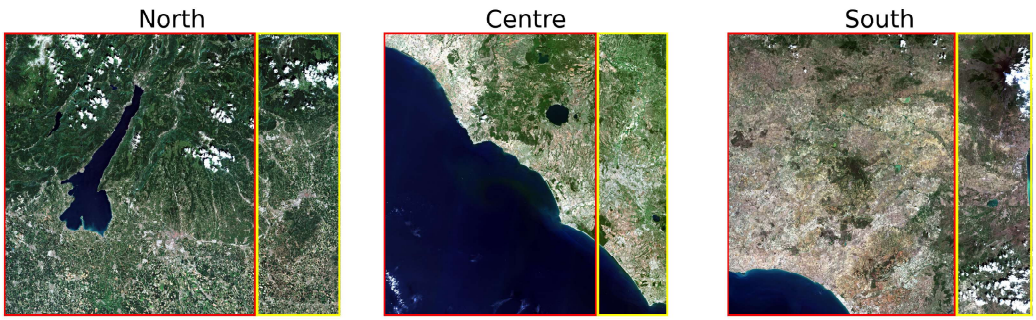


Figure 3.2: Optical images at t_0 for North, Centre and South regions (from left to right). The training area is in red, while the test one in yellow.

class built from PyTorch utilities: SAR data are made of a single channel with Alpha values, while colored optical images consist of three channels each to account for RGB values. Since the algorithm works with tensors, the input is a tensor of size nx128x128[1] (n is 7 for one optical and four SAR images); the ground truth, as well as the output, are of size 3x128x128 since they are color images. All the inputs are scaled in the range $[-1, 1]$ before passing through the networks, as for [30].

3.3 Network architectures

As discussed before, a cGAN consists of two networks: the generator (figure 3.4) and the discriminator (figure 3.5).

The generator consists of three ResNeXt blocks and six convolutional blocks. The ResNeXt blocks are used instead of standard ResNets as for [28]. They are composed with Convolution-BN-ReLu-Dropout-Convolution-BN (figure 3.3); only three blocks were added to decrease the number of parameters and allow the network to run on limited hardware. The convolutional blocks are made of Convolution-BN-ReLu. The BN layer is used to avoid the vanishing gradient problem and is suggested in [30], [39]. ReLu is used to increase non-linearity [11]. Dropout layers are

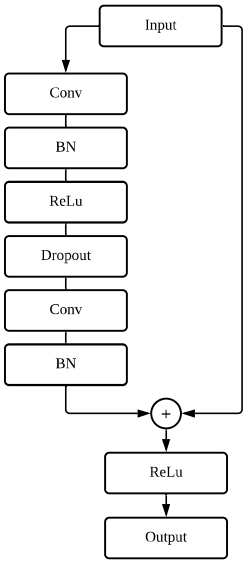


Figure 3.3: Framework used for the residual blocks.

[1]Number of channels x Height x Width

Table 3.2: Setting for Generator network.

Block	Features	Kernel	Stride	Padding
Block1	64	7	1	3
Block2	128	3	1	1
Block3	256	3	1	1
ResNeXt blocks	256	3	1	1
Block4	128	3	1	1
Block5	64	3	1	1
Block6	3	3	1	1

used to increase the robustness of the network
and help in achieving good results in case random noise is not added as input [11], [40]. The last layer consists of Convolution-Tanh, which helps to ensure that pixels in the output image are in the proper value range, as for [30]. The setting of each block (features number, kernel size, stride and padding) is shown in table 3.2: they are set to keep the spatial size of input and output images and avoid spatial reduction; for the same reason pooling steps were removed. The generator takes a seven-layer tensor as input, increases the number of features at the beginning, keeps them constant through the residual blocks, and decreases them to output a three-layer tensor that works as an RGB image.

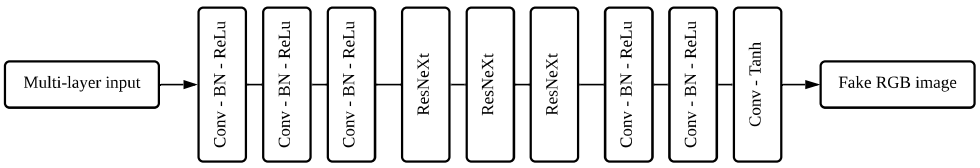


Figure 3.4: Architecture for the Generator network.

The architecture for the discriminator, instead, is borrowed from [11], [40] and is called patchGAN: this solution is useful to capture high-frequency structures and runs faster due to reduced parameters. The network is made of five convolutional blocks: the first one is composed of Convolution-LReLu, the last one performs Convolution only, while those in the middle are in the form Convolution-BN-LReLu. The BN layer is used for the same reasons as before. The *Leaky ReLu* function is used here rather than ReLu, as for [30], [41], [42]. The settings for the blocks are shown in table 3.3: they are set to increase the number of features through the network while decreasing the spatial size of the image. Thus, the input is an RGB image as a three-layer tensor and the network outputs a single-layer tensor of size 14x14 with values in the range $[0, 1]$. This image is used

Table 3.3: Setting for Discriminator network.

Block	Features	Kernel	Stride	Padding
Block1	64	4	2	1
Block2	128	4	2	1
Block3	256	4	2	1
Block4	512	4	1	1
Block5	1	4	1	1

to classify the input as real or fake during training. So, the network assigns a probability to be real or fake to a smaller patch instead of to the whole image.

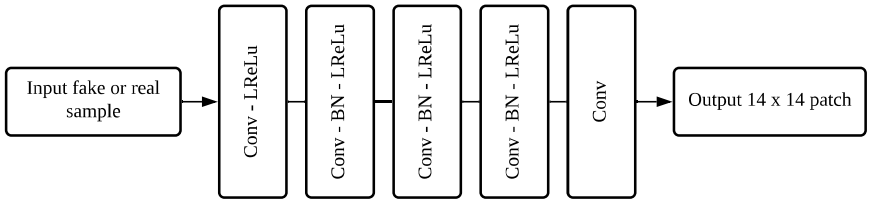


Figure 3.5: Architecture for the Discriminator network.

All weights of the networks were initialized from a zero-centered Normal distribution with standard deviation 0,02. When the *Leaky ReLu* function is used, the negative slope is set to 0,2 [30].

3.4 Training and evaluation

The machine learning part was coded in Python, using PyTorch and TorchVision libraries for dataset creation, network building, training and testing. The adversarial process is modeled as a minimax game between the generator and the discriminator, so the objective function is defined as:

$$\begin{aligned}\min_G \max_D \mathcal{L}_{cGAN}(G,D) = \mathbf{E}_{(x,y)\in p_{data}(x,y)}[\log(D(x,y))]+ \\ +\mathbf{E}_{x\in p_{data}, z\in p_{data}(z)}[1-\log(D(x,G(x,z)))]\end{aligned} \tag{3.1}$$

This function comprises the total loss of the model. However, as for [11], [40], the objective of the generator is not only to fool the discriminator, but also to produce an output similar to the ground truth: to accomplish this task and encourage less blurring, the L1 distance is absorbed into the initial loss.

$$\mathcal{L}_{L1} = \mathbf{E}_{(x,y)\in p_{data}(x,y), z\in p_{data}(z)}\|y - G(x,z)\|_{L1} \tag{3.2}$$

So the objective function is:

$$G^* = \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{3.3}$$

To achieve this objective, the Binary Cross Entropy loss function is used for the main part; for the L1 distance, the L1 loss function is used instead. Regarding the BCE loss, it is defined as:

$$l(x, y) = L = \{l_1, \ldots, l_N\}^T, \quad l_n = -w_n[y_n \cdot \log(x_n) + (1 - y_n) \cdot \log(1 - x_n)] \tag{3.4}$$

This function allows us to calculate both the terms of the main objective function. Since it is a binary classification problem, it is possible to specify what part of the loss function to use by changing y (i.e., the ground truth which takes values 1 or 0, so it permits to select which term to use) [43]. Furthermore, backpropagation is used to get the gradient of the loss, while the Adam optimizer updates the model.

The workflow for the training of the model requires to train the generator and the discriminator simultaneously. In a nutshell, the generative network produces a fake image and the discriminative one tries to classify the input image as fake or real. In practice, the discriminator is trained first to improve its classification accuracy, and then it is used to train the generator [11]. The process iterates until the end of the training, i.e. after all the epochs.

To make this possible, the training is accomplished through separate batches for real and fake images (i.e., mini-batch technique from [39], [43]), using the BCE loss function as discussed before, in combination with the L1 loss function with regards to the generator. The mini-batch solution allows using only one loss function to train the model, since the real and fake classification is evaluated separately. The two losses are then added together to get the final value to update the network.

The training for the discriminator follows the workflow:

- Load real sample
- Forward it through the discriminator
- Calculate the loss for real images
- Get the gradient with a backward pass
- Create a fake sample by forwarding the input sample through the generator

- Forward pass through the discriminator
- Calculate the loss for fake images
- Accumulate the gradient with a backward pass
- Optimize the network

Following in the training loop, the generator is trained afterward:

- Forward pass the fake sample through the trained discriminator
- Calculate the loss for the fake image (in this case, the ground truth is the real label, since the generator works in opposition to the discriminator)
- Get the gradient with a backward pass
- Get the L1 loss between the fake sample and the real one
- Accumulate the gradient with a backward pass
- Optimize the network

The testing phase followed and was used to evaluate the accuracy of the trained generator. In this phase, the PyTorch-metrics library from Hiroaki Go and Chokurei was used.

In practice, real samples from the test dataset were loaded and passed through the generator. Then, the fake images were compared with their real counterpart (i.e., the most recent satellite image) according to two evaluation indicators:

- Peak Signal-to-Noise Ratio (PSNR)
- Structural Similarity (SSIM)

3.5 Implementation settings

The noise is absorbed into the dropout layers and not in the input x [11], [40]: for this reason, the input to the generator has 7 channels. The dropout rate is set to 0,5. The Adam optimizer parameters are 0,5 for β_1 and 0,999 for β_2 [11], [43]. The batch size is 1, and the learning rate is set to 0,0002. The coefficient λ for the L1 loss is set to 100 [44] to help accuracy and sharpness [11].

3.6 Tests setup

Two main tests were conducted to accomplish the objective of this work.

Test 1 The purpose of the first experiment is to prove the functionality of the model and its effectiveness on different datasets. Three new models were trained, each of them on a different region. Then, their training and performances were compared to understand how differences in the dataset affect the results.

Test 2 This case proposes to evaluate how the training on different datasets can affect the performance of a model. This time, a pre-trained model is trained on a different dataset: the model trained on the North dataset is trained on the Centre one; then, it was trained again on the South dataset. The performance of the single-dataset model and these two new models were compared to evaluate the effectiveness of this method.

In both cases, generated patches of size 128x128 regarding the whole source images were recomposed to compare the overall result of the network and find a possible way to return to the satellite image.

3.7 Results

The training was completed on a single GTX1070OC Graphical Processing Unit (GPU) built in a laptop; it lasted 100 epochs with a training dataset of 500 patches and 200 for the test dataset. All the regions had the same size of the dataset.

The models were trained and tested on the patch datasets; furthermore, the source image was recomposed as a final test for the networks. The models are named after the region in which they were trained.

The performance indices were evaluated between test patches as well as between simulated and original satellite images. The performances regarding the whole images were evaluated and shown in table 3.4. The indices for each region regarding the original image are shown in parenthesis and are used as a baseline for the other sets: they represent the distance between the two original optical images, so are used as comparison indices. For a visual and qualitative evaluation of Test1, it is possible to compare the first and the last row in figure 3.17. Data from training and testing are compared afterward.

Table 3.4: Indices values regarding the whole images for Test1.

	North	Centre	South
PSNR	12,22 (11,56)	11,76 (19,51)	13,584 (16,12)
SSIM	0,4478 (0,4770)	0,2903 (0,7541)	0,4179 (0,5854)

First of all, the network can simulate a brand-new image which resembles the objective image, since it is easy to understand that the predicted image is related to the source one. This result is straightforward since the network is programmed to learn a defined distribution that lies across the dataset. On the other hand, the effectiveness is fair and is still an open problem for complex algorithms [11], [45].

It is noteworthy that the hardware works a pivotal role in machine learning tasks [46]. Satellite images are usually large and heavy, and require sufficient memory to manage them. Then, deep learning in computer vision tasks is computationally expensive since complex networks are used together with massive data with a large distribution. Very deep networks require many more parameters, but become necessary when dealing with high features learning. In this case, the residual blocks in [11] were reduced from nine to three, and the train and test datasets were set at 500 and 200 patches, respectively, with images of size 128x128. These decisions have decreased the amount of learning available to the network since half of the distribution was used with respect to [11]. However, the algorithm has made a leap towards image simulation: the higher indices of [11] demonstrate the relevance of better settings to achieve more effective results, but they need proper hardware support; from this point of view, GPUs are essential for hardware-accelerated tasks.

Regarding the overall performance, the indices in table 3.4 are made by comparing the whole simulated image with its real counterpart. These values are compared with each other and with the baseline. Each model tried to replicate its objective image. According to table 3.4, only the North model received a PSNR index higher than tits baseline, but a slightly lower SSIM index. However, this result is not so meaningful, since the baseline for the North region is affected by a change of information related to clouds and image colors (figure 3.18). The Centre region got the lowest indices, a symptom that a wide area of absence of feature (the sea composes half of the image) drastically weakens the effectiveness of the network; the same issue happens with clouds, as addressed in [47]. In general, all the values are quite close and lower than the evidence in [11]. Lower performance is related to the following factors:

- A small dataset with respect to the number of features to learn;
- A smaller network with the same channels has lower parameters to recognize the same amount of features;
- The dataset has to be selected appropriately.

About the last point, vast areas without features can be potentially misleading in the pursuit of the given objective. This is especially true when dealing with optical images: areas with seas or oceans, or rather clouds are not suitable for training; furthermore, daylight conditions might change the color distribution. It is thought that clear imagery caught on different timeframes may help in achieving better results because the network avoids learning misleading features in favor of more valuable ones. Indeed, the simulated images show the same pattern as the input ones for clouds and water bodies (figure 3.18). Clear imagery helps the model in learning features that might be covered in future timeframes. As highlighted by [11], radar images are important for passing the changes of the images without taking into account obstacles like clouds, since SAR images are not sensitive to them; this is true with the hypothesis that the model has previously been trained without interferences. In this case, simulated images still have clouds similar to the optical input because the model has learned their presence.

By inspecting the generated patches from the test dataset, it is possible to see how the model has learned some features and that it tries to generate the objective patch, but results are still blurred in the best cases. At this stage and with the aft-mentioned setting, it is evident that the model simulates low-features patches better than high-features ones. The top patches in SSIM values with their indices (descending from left to right) are shown in figure 3.6. They have low to nothing to simulate, but the South region patches have more features than the others, at the expense of a lower index. This means that the distribution of the model drifts towards low-features samples sooner, impacting the further training and performance; this aspect is especially true in the Centre region. On the other hand, the South region has more ground features with respect to the North (showing a lake and mountains) and the Centre (showing a wide area of sea and many green patches).

Training and test data are shown in the following pages (figures 3.8, 3.9, 3.10; figures 3.11, 3.12): training data are grouped by region and, in the first two graphs, show discriminator and generator total losses split into their parts, while

in the third one there are the values of BCE losses of both networks and split for real and simulated (i.e., fake) batches; test data are grouped by the index to compare the results from the different regions and show the average value.

All the models show divergent behavior during training. The discriminator loss for both real and fake samples decreases along the epochs; this means the discriminator becomes better and better in classifying the samples. In opposition, the generator starts stably, but its BCE loss starts increasing as the discriminator one starts decreasing. This behavior is even more evident in the third graph, since the generator L1 loss keeps stable values throughout the training. The divergence means the discriminator is defeating the generator in their adversarial game, hence the first is becoming even better in classification while the second struggles in generating valuable samples. This issue is common for GANs, since their training is particularly complex [48]: a possible solution has been found from [49] but not investigated in this work.

The test data reflect the indices regarding the whole simulated image. The South model has the lower performance, mainly because of the dataset used, while the Centre and North models got the most success, but still far from meaningful and usable results.

The importance of the dataset and the training is further investigated in the second test. The North model is trained again using the Centre dataset, and then the South one: for the purpose of this paper, the two new models are called Transfer 1 and 2, respectively. Training and test data are shown as before (figures 3.13, 3.14; figures 3.15, 3.16). Recomposed images for the three regions made from the two new models are compared as visual evaluation.

The training behavior is similar to the results from Test1. There is also a slight reduction in the generator L1 loss, meaning that fake samples are more similar to their original counterpart. However, the behavior of Transfer1 is interesting, since it seems like all BCE losses reverse their trend to attain a more stable pattern after about 25000 iterations; further epochs are needed to discover the following trend. This pattern means that the generator has found a way to generate samples that make the discriminator's job harder. This conclusion is not entirely valid, since the large number of sea-samples has led the generator to prefer meaningless blue samples to high-features ones from the ground (figure 3.7). This shift demonstrates the ability of the network to adapt, but also the pivotal role of the dataset. Lower indices values are another proof that shows the difficulty of the model in simulating an unseen distribution with no sea involved (the test area

Table 3.5: Indices values regarding the whole images for all the tests.

PSNR - SSIM	North	Centre	South
From main models	12,22 - 0,4478	11,76 - 0,2903	13,58 - 0,4179
Trasnfer 1	10,12 - 0,3455	10,90 - 0,5355	8,80 - 0,2054
Transfer 2	12,08 - 0,3462	14,71 - 0,2755	15,46 - 0,6244
Objective image	11,56 - 0,4770	19,51 - 0,7541	16,12 - 0,5854

in figure 3.2 is on purpose). On the other hand, Transfer2 model achieved better indices values for the Centre and South regions. This evidence demonstrates that more training[2] on new datasets has a beneficial effect in improving performance, because the model distribution embraces new features. The improvement of the model through time is also showed by the indices values of the simulated image from the South region for Transfer2: the PSNR is the highest of the test and the SSIM is higher than the objective image (table 3.5).

3.8 Conclusions

This thesis investigated the possibility of Sentinel-1 and -2 imagery data fusion through deep learning techniques, starting from the work of [11]. SAR and optical data from two timeframes were used as datasets to train a Conditional Generative Adversarial Network; hence, the objective was to simulate optical data starting from multi-temporal SAR-optical images combined. The tests regarded the functionality of this system and how the input dataset can change the results of the network.

Firstly, the method used can successfully generate an optical image that resembles the reference optical image and the network is valuable, despite the hardware limitations. Secondly, changes in the source dataset can drastically affect the results, even though more training alleviates the drawback of an imperfect dataset and increases the performance.

Although the model works, there is much room for improvement. The hardware becomes an essential component when working on computer vision tasks and allows for more complex and deeper models that, in turn, extend the achievable results; researches on the network can be aimed at a more stable training behavior and more robust network designs.

Furthermore, it is thought that the dataset works a pivotal role in this kind

[2]It is possible to train a network more times, not only when it is built. The process is also called data augmentation and is also related to transfer learning techniques, particularly used to adapt trained models on new jobs.

of task. The issue is twofold: the dataset quality might be addressed by a careful selection, while its size should be optimized according to the features of the objective image, both in extension and in time. In this way, data augmentation or transfer learning techniques might be useful to increase the amount of data available or exploit pre-trained models.

Further improvements can also be aimed to complete the pipeline for this thesis work with autonomous solutions to create a fully-operable system.

The Github repository for this project is at `https://github.com/Matthew2595/fusion_cgan`.

3.9 Defense applications

This thesis derives from the need to investigate possible capabilities that may be interesting for the Defense Department, with regards to remote sensing imagery, especially with the recent increasing interest in dual-use satellite systems [50]. These systems may be beneficial for State actors, without renouncing civilian usage; on the other hand, they become tools and targets of strategic importance [51].

As for [52], satellites are used by the military for a variety of fields: communications, navigation, remote sensing. The last one is of particular interest and invaluable strategic and tactical support; it helps in:

- Gathering data from multiple spectra, allowing the process of new and augmented information [53], [54];
- Intercepting radio or data transmissions with intelligence, surveillance and reconnaissance advantages [55];
- "Strategic planning, deployment, targeting, and threat assessment" [56], [57].

Overall, it is a tool in augmenting knowledge either on the battlefield or at home.

Furthermore, Copernicus has a Security Service among the others [58], and it represents an example of a dual-use satellite system: services are delivered to private end-users and State members (EU included) with different purposes. In this case, the security encloses border and maritime surveillance, and support to EU external action. This example shows the potential of such an integrated system for activity report, support to evacuation plans, infrastructures assessment,

ground and camp analysis, crisis and conflicts pictures for support to humanitarian actions, border security and stability development [58]. The service is accessed via a request submission and approval, and the delivery of products is secured. Since it is a space-based system, the coverage is worldwide, with clear benefits.

In this outlined environment, it is clear to see how this thesis work can benefit some established tasks or create new ones.

Data fusion can merge similar and dissimilar information to attain a more comprehensive result: different spectra can be merged as needed to back up decision-making processes and tactical actions with an enhanced awareness of the environment. As for this work, changes in visual information can be accessed only via some wavelengths, thus helping in completing the future data with unseen changes [59]–[61].

The image simulation task may help in dealing with cloud covers in optical imagery, or rather preventing or generating new data for regions with missing or no data at all. Again, simulated images can help if compared with real data to enhance changes evaluation, a critical task for decision-making actors. Another related task is Super Resolution, where new images are generated with increased resolution and information, essential for target reconnaissance, acquisition and assessment [62]–[64].

In these cases, military or dual-use satellites are able to improve the potential results thanks to higher resolution imagery, hyperspectral capabilities and access to classified information. These three elements boost the creation of more valuable products by increasing the amount of data available as input and, hopefully, not available to the adversary. This particular aspect poses the attention to the importance of aerial and space superiority. Moreover, faster and classified communication systems help in delivering data and products to all the actors in real-time or near real-time. This feature is of particular interest in critical-decision environments.

As the last point, the added value of machine learning techniques cannot be understated. Autonomous solutions of this kind act as a glue for the previously discussed ideas and create new tasks. Firstly, image classification tasks can be improved using neural networks since they help to understand underlying patterns of data [65], [66]. Then, generative networks can be used for simulation and super-resolution tasks. Finally, image processing encloses several tasks: the cited evaluation of changes, targeting, interferometry, object recognition, segmentation,

captioning, debiasing and outlier detection. Some of them can also be reused for the creation of more complex models.

The evolution of Artificial Intelligence affects hardware development, too. The discussed tasks can be parallelized, creating multiple instances that deliver real-time products to end-users or to other processes, with the idea to create completely autonomous customized machines. In this way, there are many paths to research for new and advanced solutions.

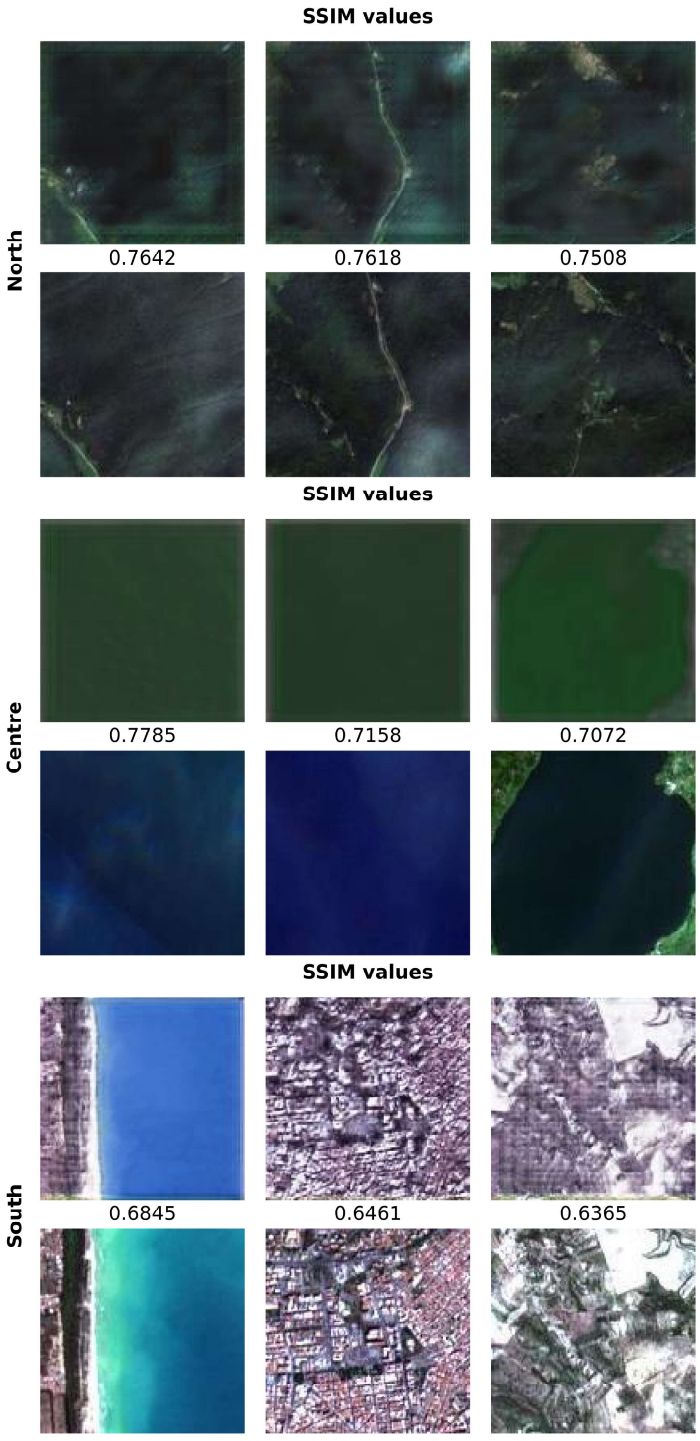


Figure 3.6: Top 3 patches in SSIM during training per region. The index is descending from left to right, and the top rows contains the simulated patches.

Figure 3.7: Four samples around 25000 iterations. The first two on the left are before the adaptation of the network. The model preferred blue samples the improve its performance against the Discriminator, but the results are visually worse.

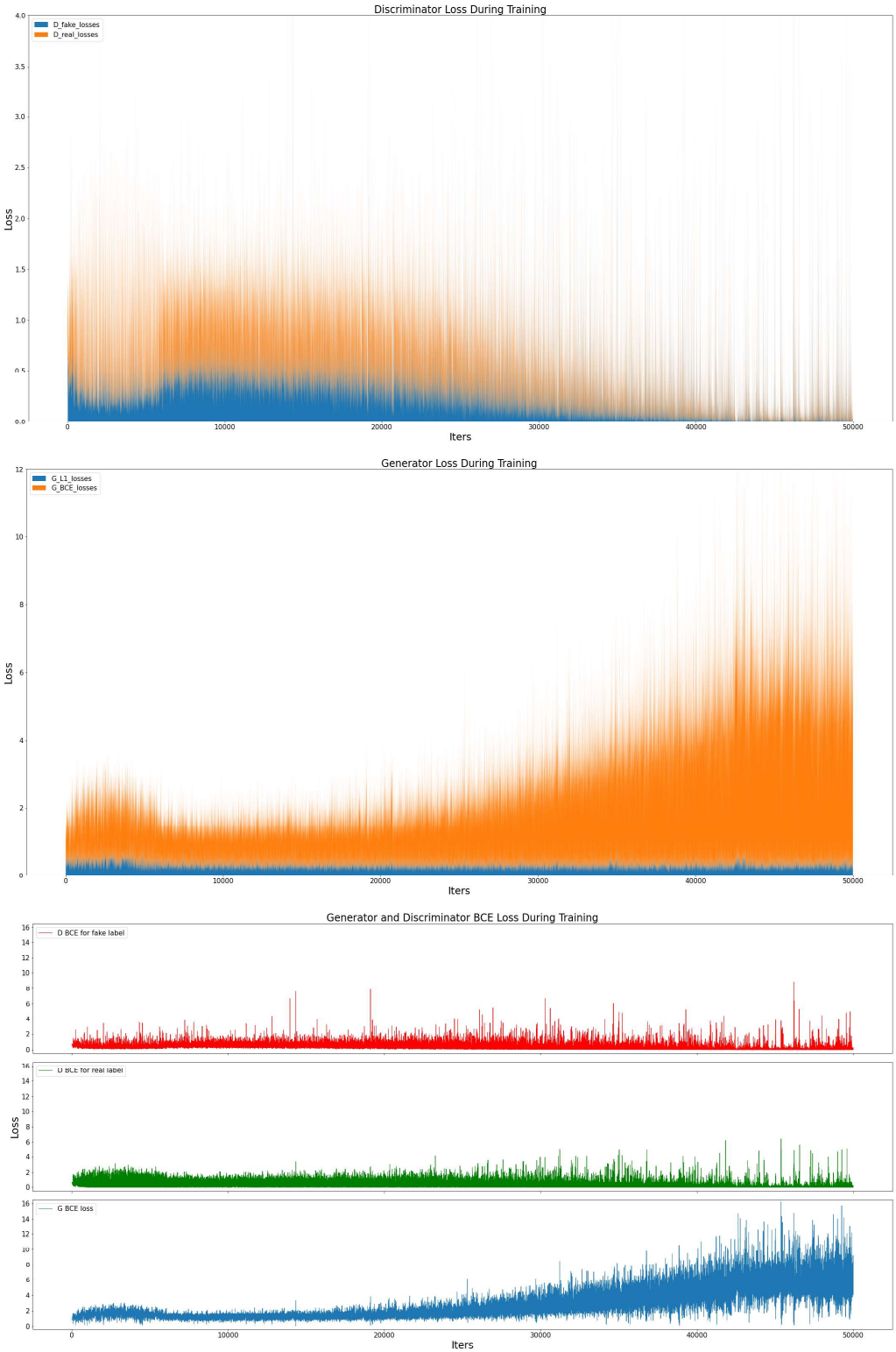


Figure 3.8: Training data regarding the North model during Test 1.

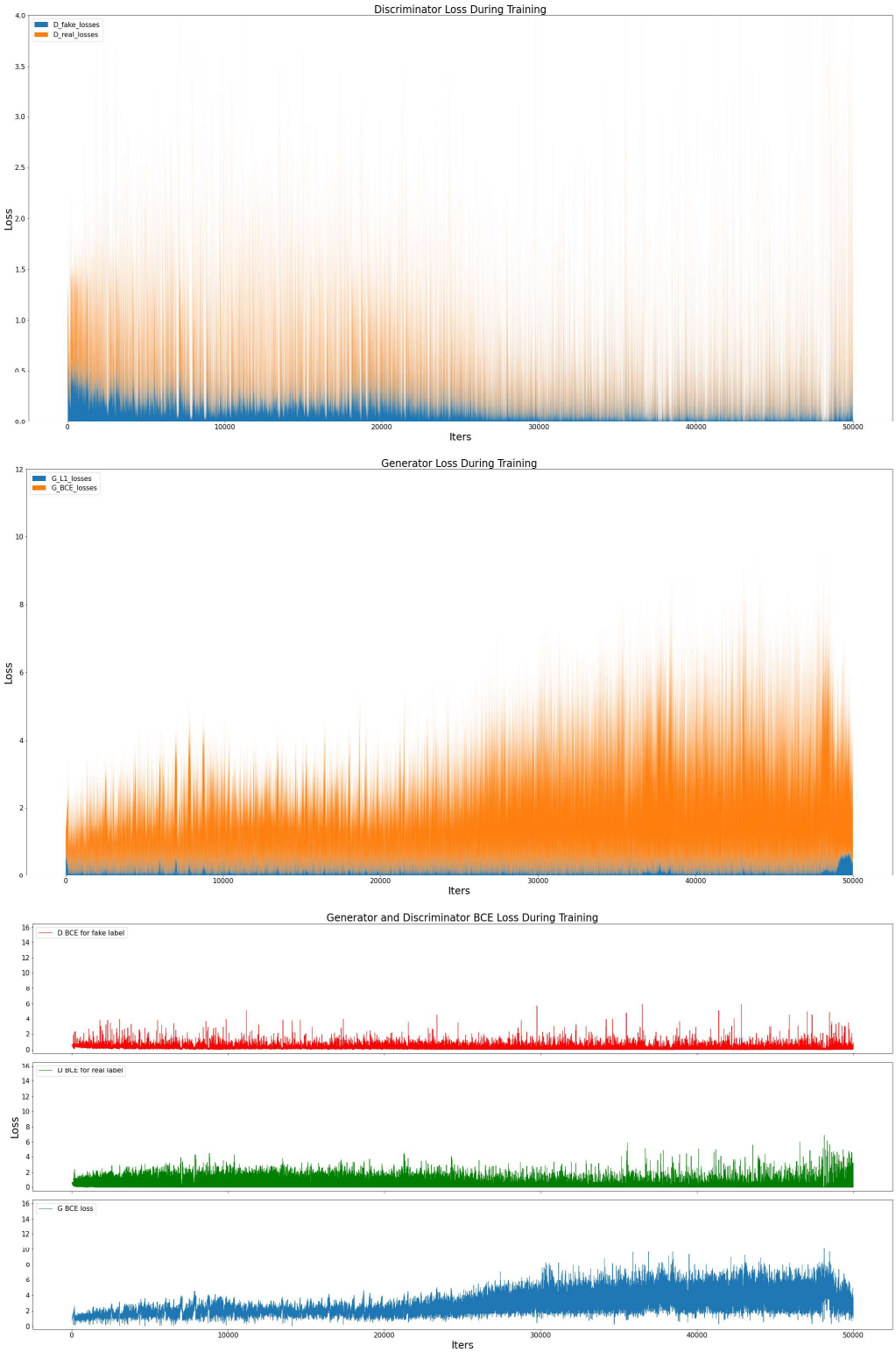


Figure 3.9: Training data regarding the Centre model during Test 1.

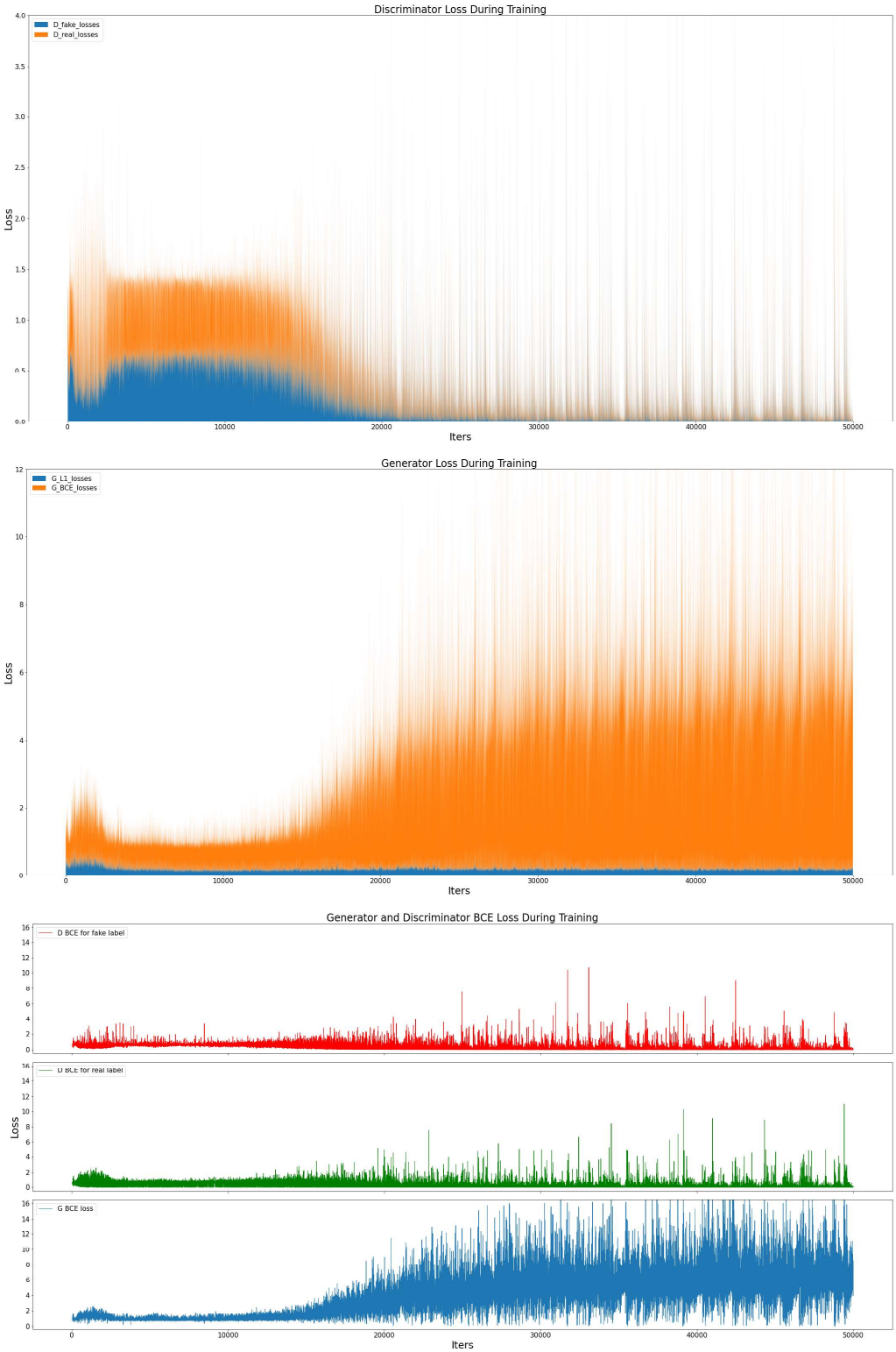


Figure 3.10: Training data regarding the South model during Test 1.

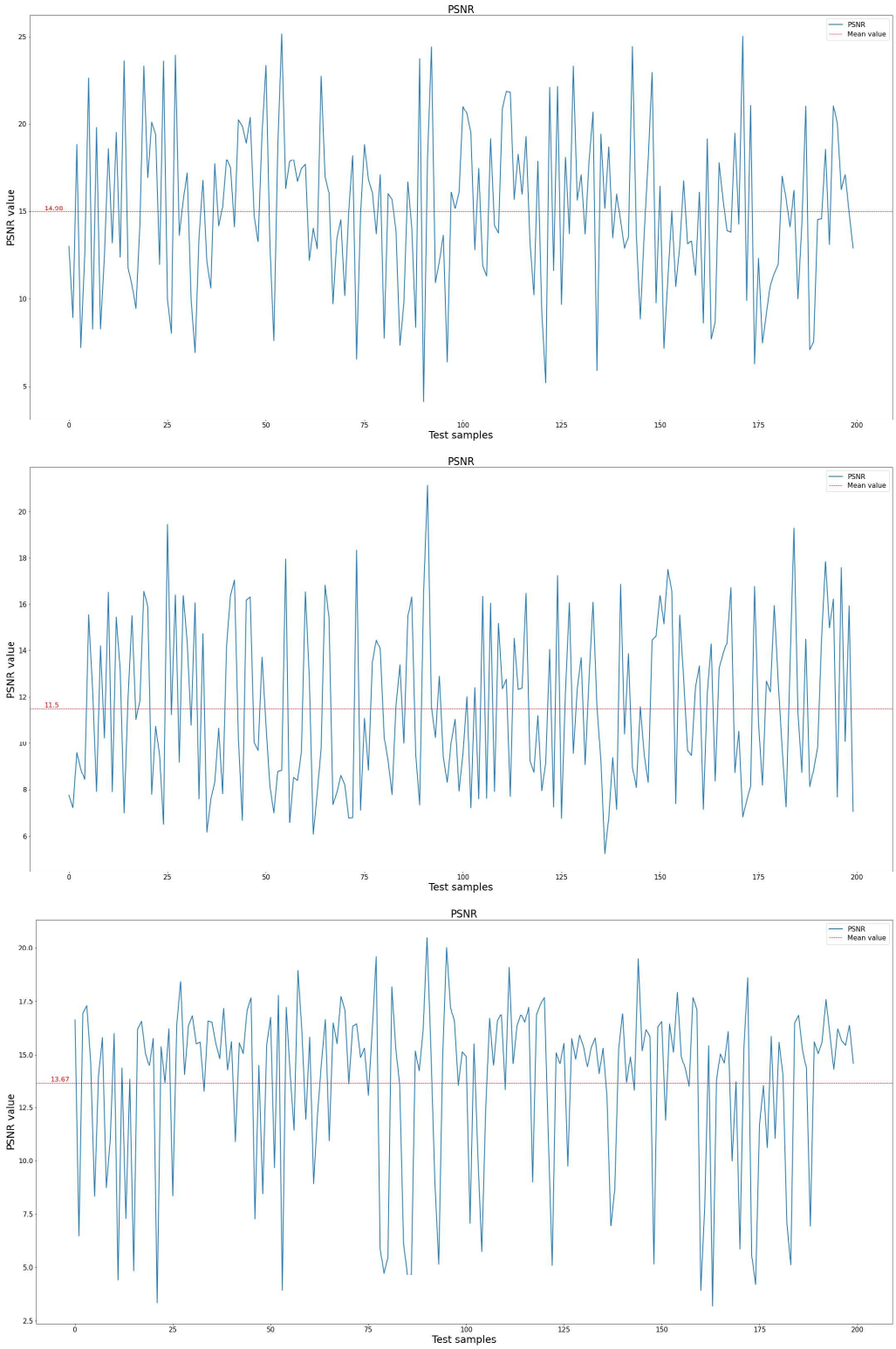


Figure 3.11: PSNR values from all the regions during Test 1.

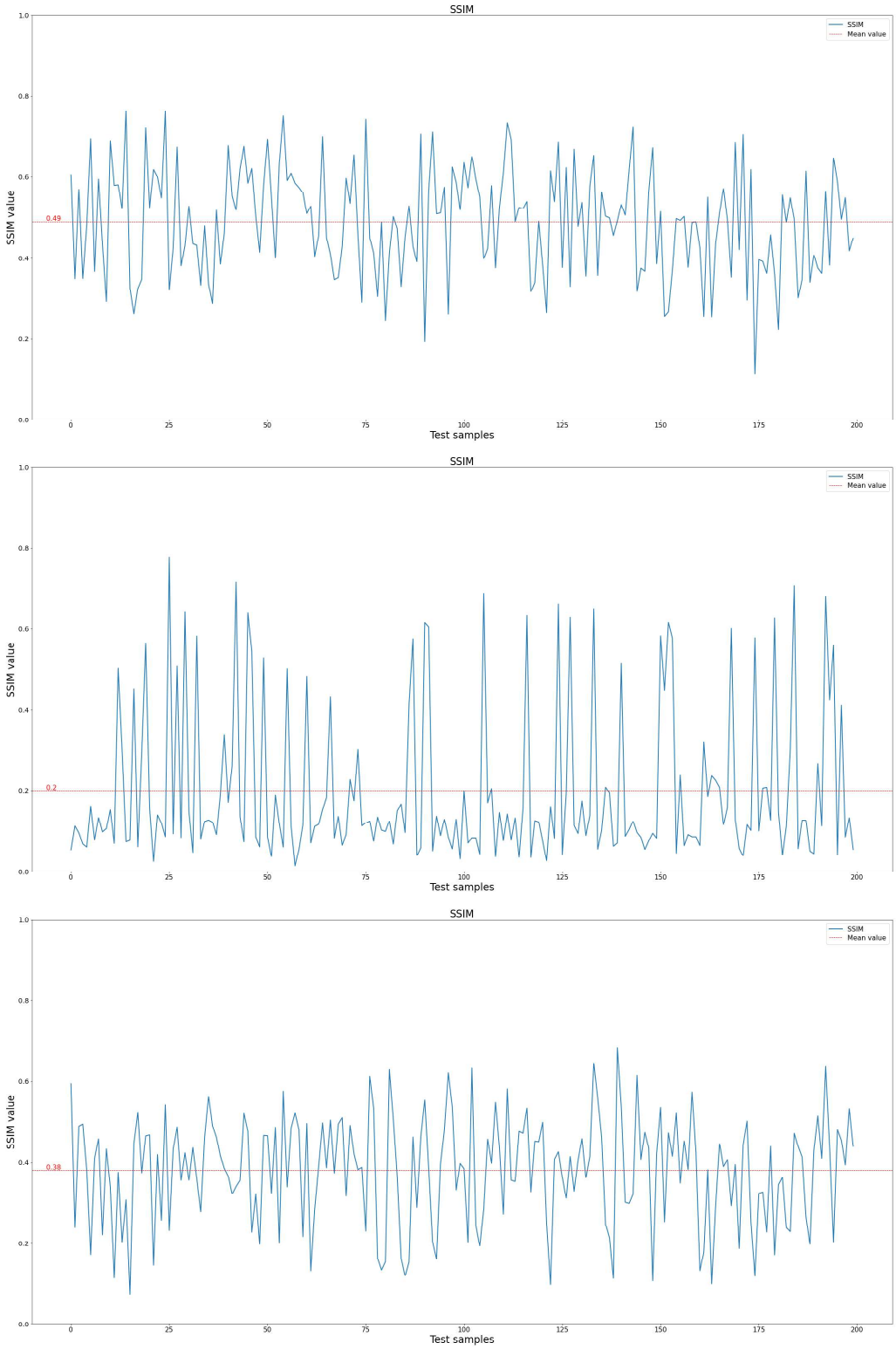


Figure 3.12: SSIM values from all the regions during Test 1.

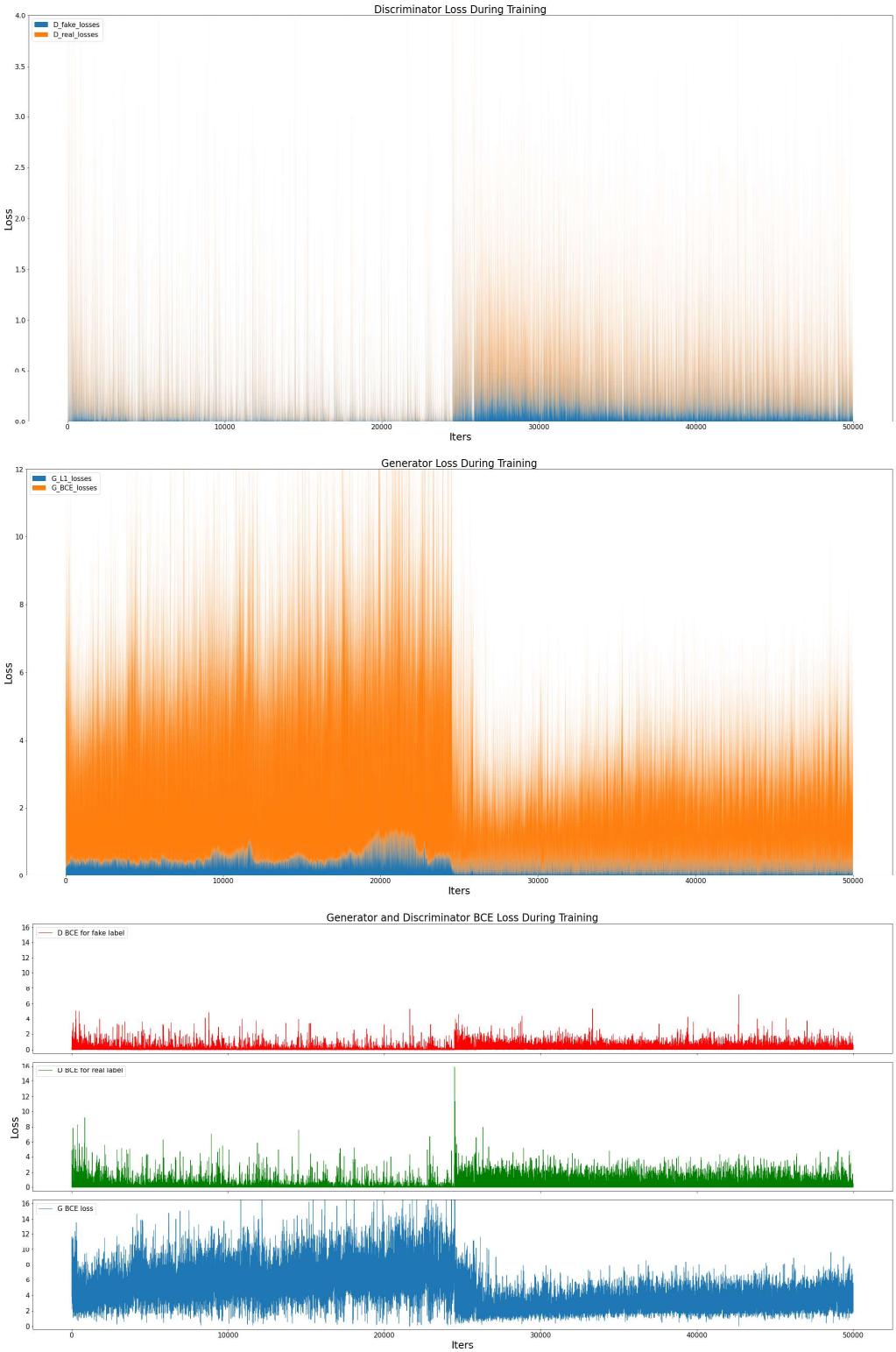


Figure 3.13: Training data regarding the Transfer 1 model.

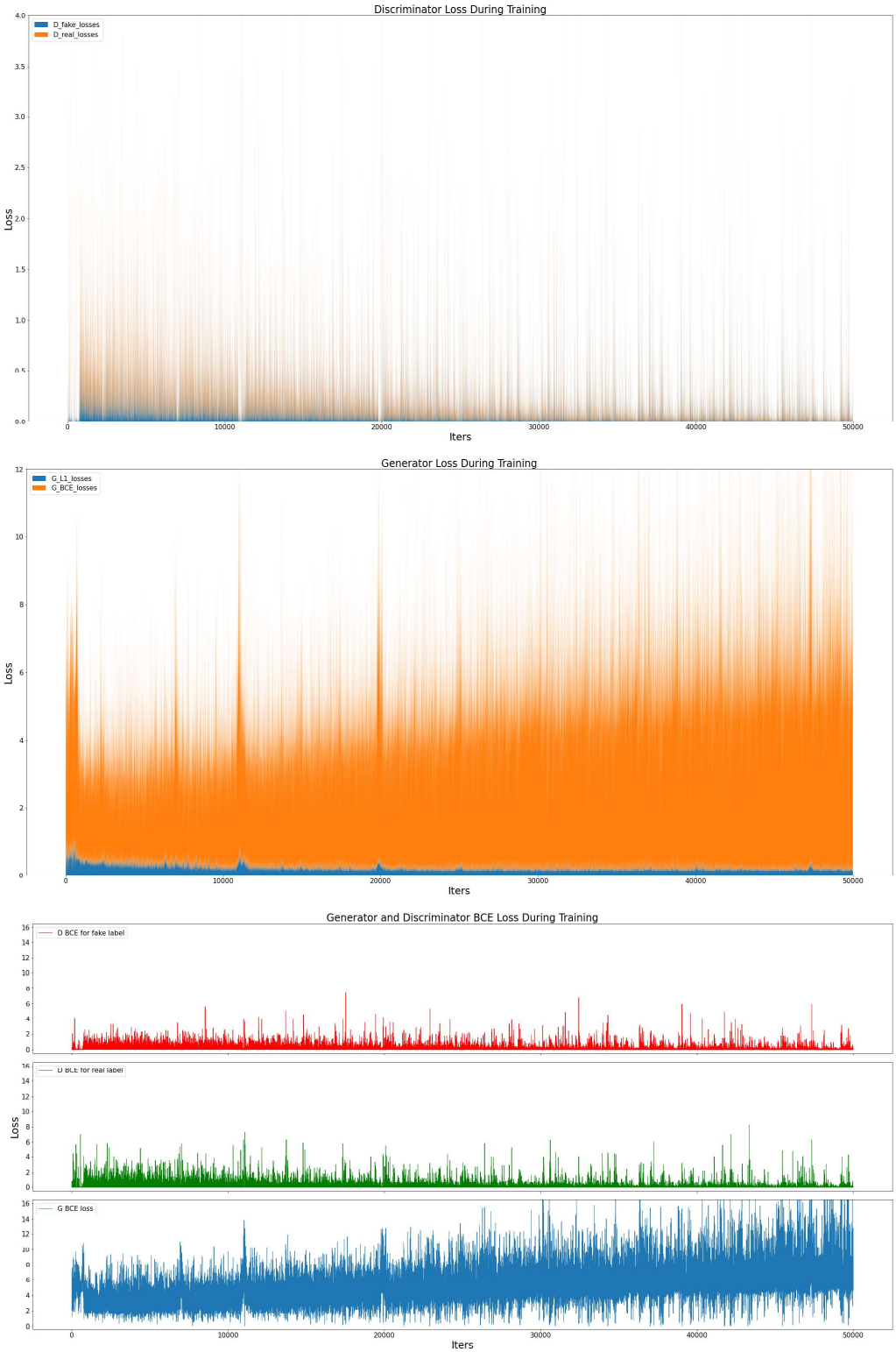


Figure 3.14: Training data regarding the Transfer 2 model.

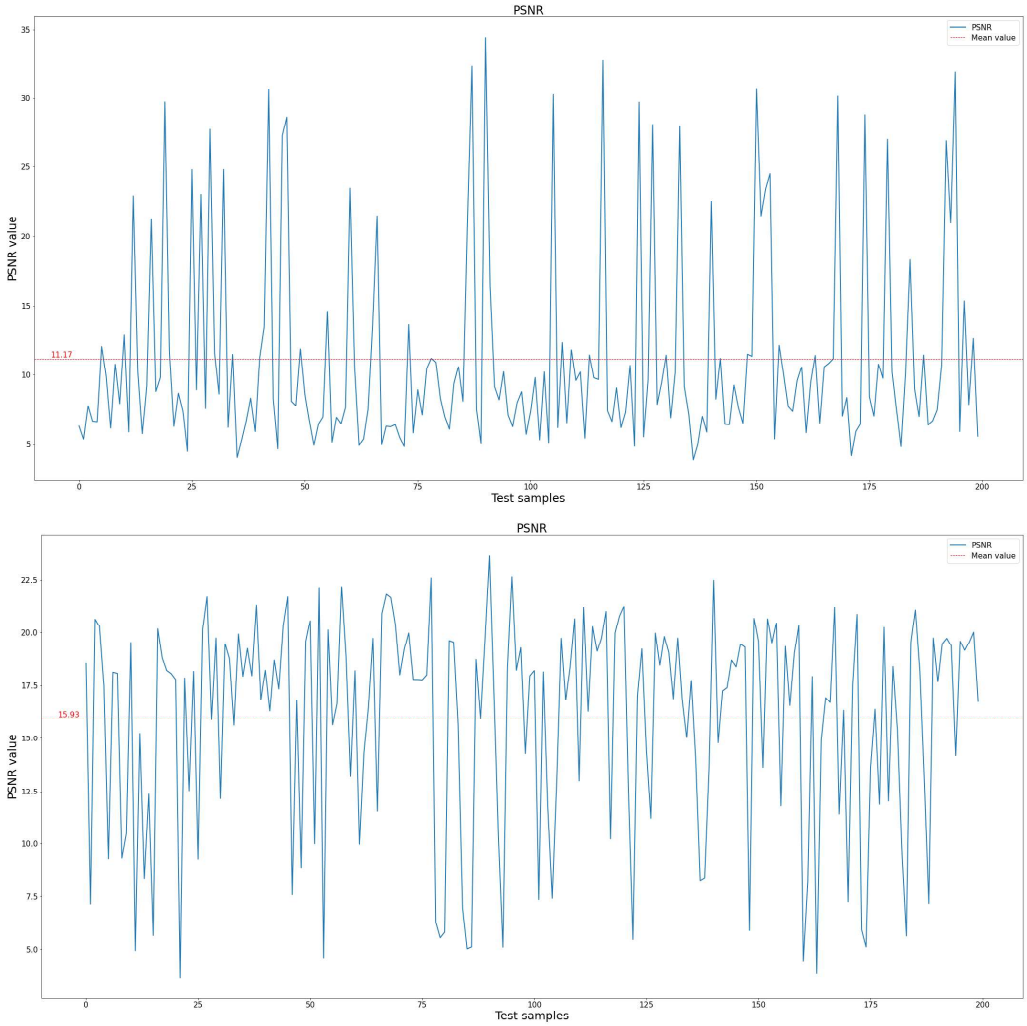


Figure 3.15: PSNR values from all the regions during Test 2.

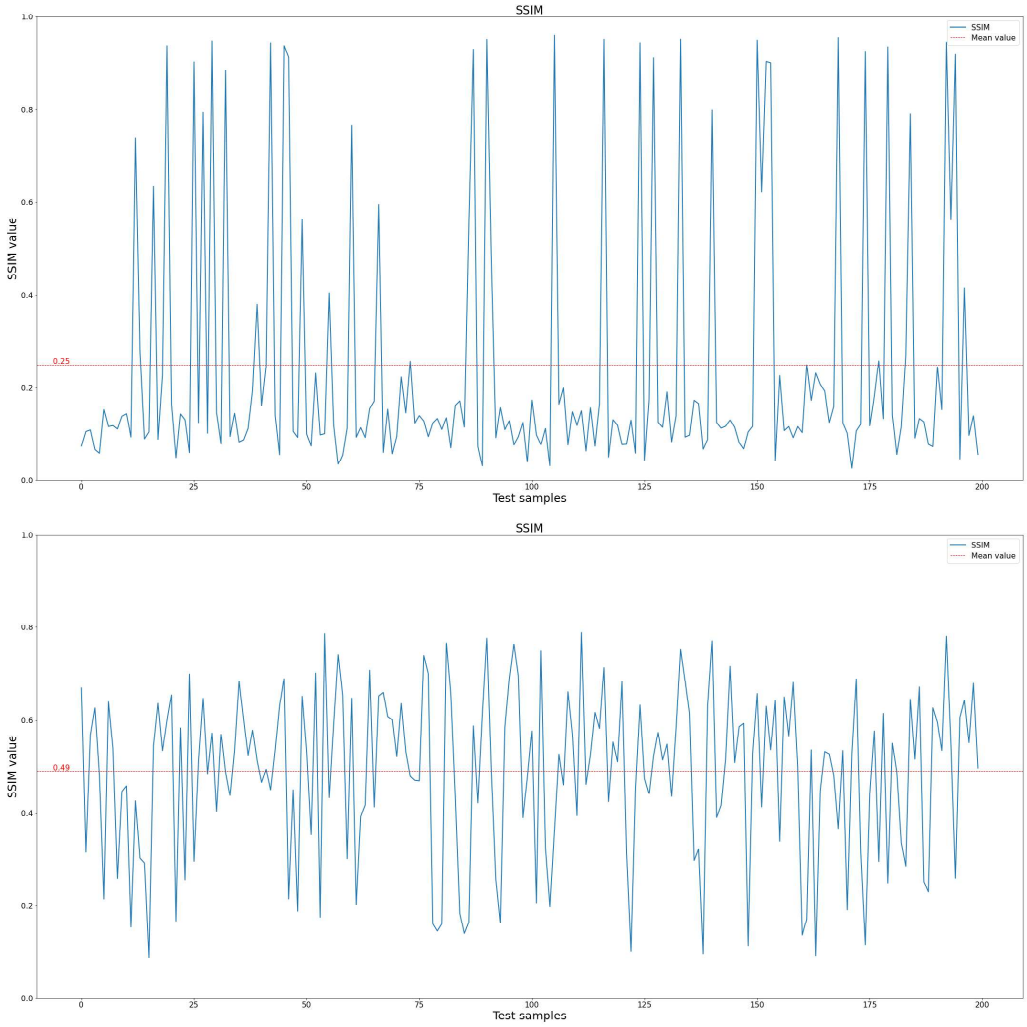


Figure 3.16: SSIM values from all the regions during Test 2.

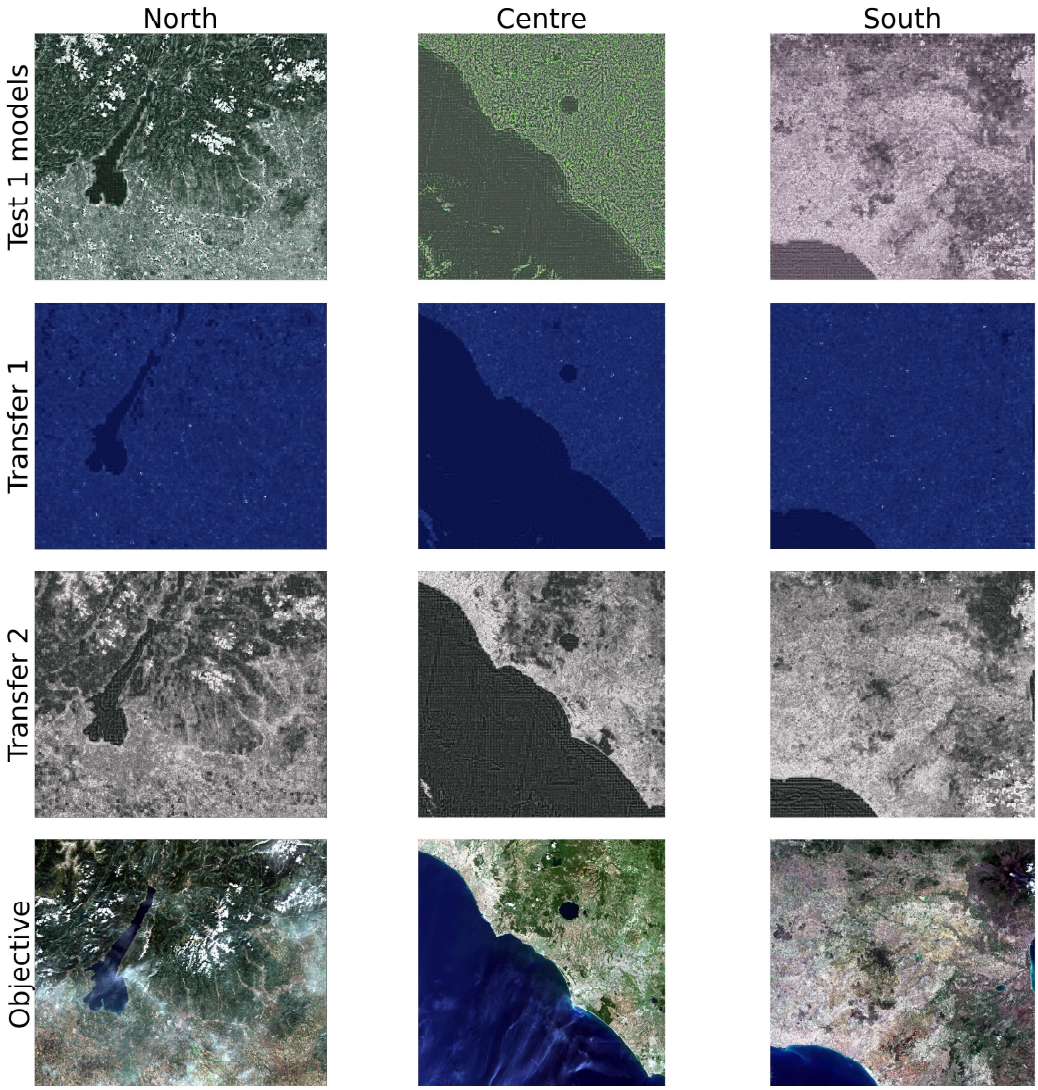


Figure 3.17: Visual results from all the tests. First row shows simulated images from the first Test. Second and third rows show simulated images after the second Test, while the last row reports the objective image to simulate.

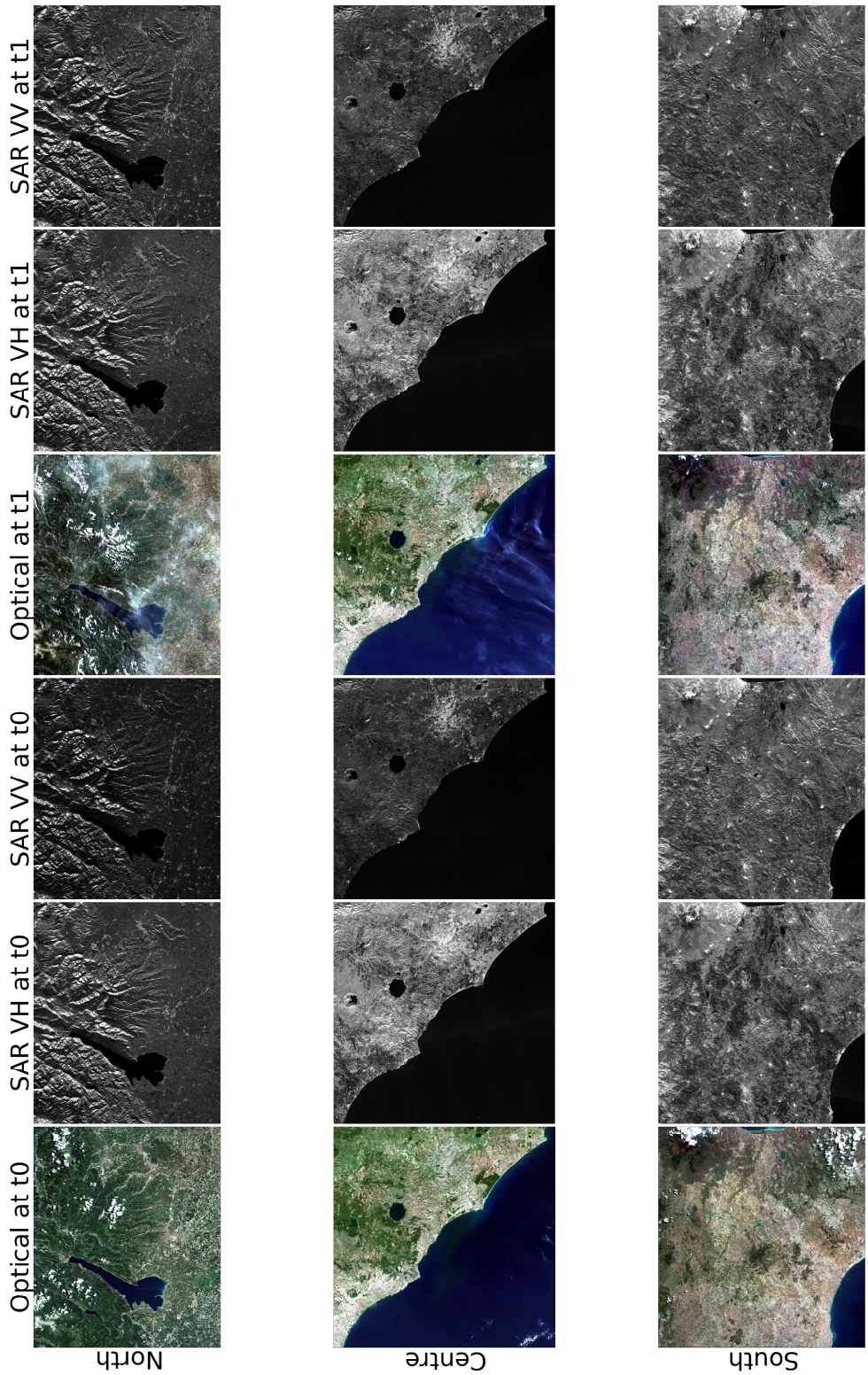


Figure 3.18: All source images used to create the needed datasets.

Acronyms

AOCS Attitude and Orbit Control System. 17

BCE Binary Cross Entropy. 33, 44, 49

BN Batch Normalization. 29, 41, 42

BOA Bottom Of Atmosphere. 18

C3S Copernicus Climate Change Service. 20

CAMS Copernicus Atmosphere Monitoring Service. 19

CCF Computer Compatible Format. 14

cGAN Conditional Generative Adversarial Network. 36, 39, 41, 50

CLMS Copernicus Land Monitoring Service. 20

CMEMS Copernicus Marine Environment Monitoring Service. 19

CMES Copernicus Emergency Management Service. 20

CNN Convolutional Neural Network. 28

COTS Commercial Off-the-Shelf. 8

CRS Coordinate Reference System. 24, 25

CSAR C-band SAR. 11, 13, 14

CSM Calibration and Shutter Mechanism. 18

DEM Digital Elevation Model. 24, 25

DG Directorate-General. 20

ECHO European Civil Protection and Humanitarian Action. 20

ECMWF European Centre for Medium-Range Weather Forecasts. 3, 19, 20

EDRS European Data Relay Satellite. 14

EEA European Environment Agency. 3, 19

EMSA European Maritime Safety Agency. 3, 20

EO Earth Observation. i, iii, 1–4, 6–9, 20, 23

ESA European Space Agency. 2, 6, 8, 10, 20, 23

EU European Union. 2, 3, 6, 10, 19, 20, 51

EW Extra Wide-swath. 13, 15

FDBAQ Flexible Dynamic Block Adaptive Quantization. 13, 14

FPA Focal Plane Assembly. 18

FRONTEX Frontières extérieures. 3, 20

GAN Generative Adversarial Network. i, 35, 36, 49

GMES Global Monitoring for Environment and Security. 2, 16

GNSS Global Navigation Satellite System. 17

GNU GPL GNU General Public License. 23

GPU Graphical Processing Unit. 46, 47

GRD Ground Range Detected. 15, 23, 24

ISP Instrument Source Packets. 14

IW Interferometric Wide-swath. 13, 15

JRC Joint Research Centre. 3, 20

LReLu Leaky Rectified Linear. 32, 42

MSI MultiSpectral Instrument. 17, 18, 24

NESZ Noise Equivalent Sigma Zero. 13

NIR Near-Infrared. 17, 18

NRT Near Real-Time. 15

OCN Ocean. 15

OSW Ocean Swell spectra. 15

OWI Ocean Wind field. 15

PDGS Payload Data Ground Segment. 14, 18

PRIMA Piattaforma Italiana Multi Applicativa. 11, 12

PSNR Peak Signal-to-Noise Ratio. iv, 35, 45, 47, 50, 59, 63

ReLu Rectified Linear. 32, 41, 42

ResNet Residual Network. iii, 30, 31

RGB Red, Green and Blue. 33, 38, 40, 42

RVL Surface Radial Velocity. 15

S1TBX Sentinel-1 Toolbox. 23

S2TBX Sentinel-2 Toolbox. 23

SAFE Sentinel Archive Format for Europe. 14

SAR Synthetic Aperture Radar. i, 10, 11, 14, 15, 22–24, 38–40, 48, 50

SAS SAR Antenna Subsystem. 13

SatCen European Satellite Centre. 3

SLC Single Look Complex. 15

SM Strip Map. 13, 15

SNAP Sentinel Application Platform. i, 1, 8, 22–24, 38, 40

SPOT Satellite Pour l'Observation de la Terre. 16

SSIM Structural Similarity. iii, iv, 35, 45, 47, 48, 50, 54, 60, 64

SWIR Short Wave Infrared. 17, 18

TOA Top Of Atmosphere. 18

TT&C Telemetry, Tracking and Command. 12

UAV Unmanned Aerial Vehicle. 19

VAS Value Added Services. 7

VH Vertical-Horizontal. 38

VIS Visible. 17

VV Vertical-Vertical. 38

WV Wave. 13

Bibliography

[1] European Comission, "Copernicus - Europe's eyes on Earth", *European Commission*, p. 28, 2015. DOI: 10.2873/93104. [Online]. Available: http://europa.eu%20copernicus.eu.

[2] P. Snoeij, E. Attema, M. Davidson, B. Duesmann, N. Floury, G. Levrini, B. Rommen, and B. Rosich, "Sentinel-1 radar mission: Status and performance", in *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, 2010, pp. 32–39. DOI: 10.1109/MAES.2010.5552610.

[3] Copernicus, "Market report", *Market report*, 2019. DOI: 10.2873/011961. [Online]. Available: http://www.europa.eu%20https://www.copernicus.eu/en/news/news/observer-market-report-2019.

[4] House of Commons European Scrutiny Committee, "Space Strategy for Europe", no. February, pp. 10–11, 2017.

[5] European Union, *Copernicus Web Page*, 2019. [Online]. Available: https://www.copernicus.eu/en%20http://www.copernicus.eu (visited on 09/16/2020).

[6] R. Torres, P. Snoeij, D. Geudtner, D. Bibby, M. Davidson, E. Attema, P. Potin, B. Ö. Rommen, N. Floury, M. Brown, I. N. Traver, P. Deghaye, B. Duesmann, B. Rosich, N. Miranda, C. Bruno, M. L'Abbate, R. Croci, A. Pietropaolo, M. Huchler, and F. Rostan, "GMES Sentinel-1 mission", *Remote Sensing of Environment*, vol. 120, pp. 9–24, 2012, ISSN: 00344257. DOI: 10.1016/j.rse.2011.05.028. [Online]. Available: http://dx.doi.org/10.1016/j.rse.2011.05.028.

[7] European Space Agency (ESA), *Sentinel-1 - Overview - Sentinel Online*, 2015. [Online]. Available: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1/overview%20https://sentinel.esa.int/web/sentinel/missions/sentinel-1/overview (visited on 09/02/2020).

[8] M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort, A. Meygret, F. Spoto, O. Sy, F. Marchese, and P. Bargellini, "Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services", *Remote Sensing of Environment*, vol. 120, pp. 25–36, May 2012, ISSN: 00344257. DOI: 10.1016/j.rse.2011.11.026.

[9] ESA (European Space Agency), *Sentinel-2 - Overview - Sentinel Online*, 2019. [Online]. Available: https://sentinel.esa.int/web/sentinel/missions/sentinel-2%20https://sentinel.esa.int/web/sentinel/missions/sentinel-2/overview.

[10] European Space Agency, "Sentinel-2: Data Access and Products", Tech. Rep., 2015. [Online]. Available: https://sentinel.esa.int/web/sentinel/missions/sentinel-2/.

[11] W. He and N. Yokoya, "Multi-temporal sentinel-1 and -2 data fusion for optical Image Simulation", *ISPRS International Journal of Geo-Information*, vol. 7, no. 10, pp. 1–11, 2018, ISSN: 22209964. DOI: 10.3390/ijgi7100389. arXiv: 1807.09954.

[12] ESA, *Snap / Step*, 2017. [Online]. Available: https://step.esa.int/main/toolboxes/snap/%20http://step.esa.int/main/toolboxes/snap/ (visited on 09/13/2020).

[13] European Space Agency, *Sentinel 1 Toolbox / STEP*, 2016. [Online]. Available: https://step.esa.int/main/toolboxes/sentinel-1-toolbox/%20http://step.esa.int/main/toolboxes/sentinel-1-toolbox/ (visited on 09/13/2020).

[14] ESA, *Sentinel 2 Toolbox / STEP*. [Online]. Available: https://step.esa.int/main/toolboxes/sentinel-2-toolbox/ (visited on 09/13/2020).

[15] L. Veci, "SAR Basics Tutorial, SENTINEL-1 Toolbox", Tech. Rep. March, 2016, pp. 1–20. DOI: 10.1002/smll.. arXiv: arXiv:1408.1149. [Online]. Available: http://step.esa.int.

[16] F. Filipponi, "Sentinel-1 GRD Preprocessing Workflow", *Proceedings*, vol. 18, no. 1, p. 11, 2019, ISSN: 2504-3900. DOI: 10.3390/ecrs-3-06201.

[17] J. S. Lee, I. Jurkevich, P. Dewaele, P. Wambacq, and A. Oosterlinck, "Speckle filtering of synthetic aperture radar images: a review", *Remote Sensing Reviews*, vol. 8, no. 4, pp. 313–340, 1994, ISSN: 02757257. DOI: 10.1080/02757259409532206.

[18] Oxford University, *Home: Oxford English Dictionary*, 2005. [Online]. Available: https://www.oed.com/.

[19] A. Amini and A. Soleimany, *MIT 6.S191: Introduction to Deep Learning*. [Online]. Available: http://introtodeeplearning.com/ (visited on 09/13/2020).

[20] J. S. Perry, *IBM Developer*. [Online]. Available: https://developer.ibm.com/%20https://developer.ibm.com/es/tutorials/j-spring-boot-basics-perry/ (visited on 09/13/2020).

[21] V. Zhou, *Machine Learning for Beginners: An Introduction to Neural Networks*, 2019. [Online]. Available: https://victorzhou.com/blog/intro-to-neural-networks/ (visited on 07/11/2020).

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *32nd International Conference on Machine Learning, ICML 2015*, vol. 1, 2015, pp. 448–456, ISBN: 9781510810587. arXiv: 1502.03167.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014, ISSN: 15337928. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[26] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising", *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017, ISSN: 10577149. DOI: 10.1109/TIP.2017.2662206. arXiv: arXiv:1608.03981v1.

[27] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9906 LNCS, 2016, pp. 694–711, ISBN: 9783319464749. DOI: 10.1007/978-3-319-46475-6_43. arXiv: 1603.08155v1.

[28] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks", in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 5987–5995, ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.634. arXiv: 1611.05431.

[29] Y. LeCun, L. Bottou, G. B. Orr, and K. .-.-R. Müller, "Efficient BackProp", in, 1998, pp. 9–50. DOI: 10.1007/3-540-49430-8_2. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf.

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Convolutional Generative Adversarial Networks", Tech. Rep., 2016. arXiv: 1511.06434v2. [Online]. Available: https://arxiv.org/pdf/1511.06434.pdf.

[31] S. Sharma, "Activation Functions in Neural Networks", *Towards Data Science*, 2017. [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[32] T. M. Mitchell, *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997, p. 432, ISBN: 0070428077. [Online]. Available: http://profsite.um.ac.ir/%7B~%7Dmonsefi/machine-learning/pdf/Machine-Learning-Tom-Mitchell.pdf.

[33] *PyTorch*. [Online]. Available: https://pytorch.org/ (visited on 04/25/2019).

[34] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization", in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015. arXiv: 1412.6980.

[35] L. Prechelt, *Neural Net FAQs*. [Online]. Available: https://www.cs.cmu.edu/Groups/AI/html/faqs/ai/neural/faq.html%20http://wwwipd.ira.uka.de/%7B~%7Dprechelt/FAQ/neural-net-faq.html.

[36] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks", pp. 1–9, Jun. 2014. arXiv: 1406.2661. [Online]. Available: http://arxiv.org/abs/1406.2661.

[37] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets", 2014. arXiv: 1411.1784. [Online]. Available: https://arxiv.org/pdf/1411.1784.pdf%20http://arxiv.org/abs/1411.1784.

[38] S. Koen, *Architecting a Machine Learning Pipeline*. [Online]. Available: https://towardsdatascience.com/architecting-a-machine-learning-pipeline-a847f094d1c7 (visited on 09/14/2020).

[39] Soumith, *soumith/ganhacks: starter from "How to Train a GAN?" at NIPS2016*. [Online]. Available: https://github.com/soumith/ganhacks (visited on 09/03/2020).

[40] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks", in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 5967–5976, ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.632. arXiv: 1611.07004.

[41] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network", May 2015. arXiv: 1505.00853. [Online]. Available: http://arxiv.org/abs/1505.00853.

[42] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models", in *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[43] N. Inkawhich, *DCGAN Tutorial — PyTorch Tutorials*, 2018. [Online]. Available: https://pytorch.org/tutorials/beginner/dcgan%7B%5C_%7Dfaces%7B%5C_%7Dtutorial.html (visited on 04/25/2019).

[44] P. Ghamisi and N. Yokoya, "IMG2DSM: Height Simulation from Single Imagery Using Conditional Generative Adversarial Net", *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 794–798, May 2018, ISSN: 15580571. DOI: 10.1109/LGRS.2018.2806945.

[45] K. Sønderby, C. K. Sønderby, and L. Theis, "Instance Noise : A trick for stabilising GAN training", no. 2016, pp. 1–12, 2017. [Online]. Available: https://www.inference.vc/instance-noise-a-trick-for-stabilising-gan-training/.

[46] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The Computational Limits of Deep Learning", 2020. arXiv: 2007.05558. [Online]. Available: http://arxiv.org/abs/2007.05558.

[47] M. Schmitt, L. H. Hughes, and X. X. Zhu, "The Sen1-2 Dataset for Deep Learning in SAR-Optical Data Fusion", in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, 2018,

pp. 141–146. DOI: 10.5194/isprs-annals-IV-1-141-2018. arXiv: 1807.01569v1. [Online]. Available: www.naturalearthdata.com.

[48] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs", in *Advances in Neural Information Processing Systems*, Jun. 2016, pp. 2234–2242. arXiv: 1606.03498. [Online]. Available: http://arxiv.org/abs/1606.03498.

[49] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, "Amortised map inference for image super-resolution", in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Oct. 2017. arXiv: 1610.04490. [Online]. Available: http://arxiv.org/abs/1610.04490.

[50] V. K. Shettigara, G. M. Sumerling, and P. j. Whitbread, "Some Defence Applications Of Civilian Remote Sensing Satellite Images", Defence Science and Technology Organisation - Information Technology Division, Tech. Rep., 1993.

[51] C. Belbusti, *Remote sensing, the dual use of satellites and environment*. [Online]. Available: https://www.spacelegalissues.com/remote-sensing-the-dual-use-of-satellites-and-the-impact-on-the-environment/ (visited on 08/28/2020).

[52] R. Lee and S. Steele, "Military Use of Satellite Communications, Remote Sensing, and Global Positioning Systems in the War on Terror", *Journal of Air Law and Commerce*, vol. 79, no. 1, p. 69, 2014, ISSN: 0021-8642. [Online]. Available: http://digitalrepository.smu.edu.https//scholar.smu.edu/jalc/vol79/iss1/2.

[53] R. D. Hudson and J. W. Hudson, "The Military Applications of Remote Sensing by Infrared", *Proceedings of the IEEE*, vol. 63, no. 1, pp. 104–128, 1975, ISSN: 15582256. DOI: 10.1109/PROC.1975.9711.

[54] M. V. Persie, H. H. S. Noorbergen, a. C. V. D. Broek, and R. J. Dekker, "Use of Remote Sensing Imagery for Fast Generation of Military Maps and Simulator Databases", *Simulation*, vol. XXXIII, pp. 573–581, 2000.

[55] S. R. Stuger, "Space Based Intelligence, Surveillance, and Reconnaissance Contribution to Global Strike in 2035", AIR WAR COLLEGE, Tech. Rep., 2012. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/1018137.pdf.

[56] K. Laygo, T. W. Gillespie, N. Rayo, and E. Garcia, "Drone Bombings in the Federally Administered Tribal Areas: Public Remote Sensing Applications for Security Monitoring", *Journal of Geographic Information System*, vol. 04, no. 02, pp. 136–141, 2012, ISSN: 2151-1950. DOI: 10.4236/jgis.2012.42018. [Online]. Available: http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jgis.2012.42018.

[57] J. J. Shroder, "Remote Sensing and GIS as Counterterrorism Tools for Homeland Security: The case of Afghanistan", in, Springer, Dordrecht, 2008, pp. 11–33. DOI: 10.1007/978-1-4020-8507-9_2. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4020-8507-9%7B%5C_%7D2.

[58] European Space Agency, *Security Service Copernicus*. [Online]. Available: https://www.copernicus.eu/en/services/security (visited on 08/29/2020).

[59] B. Pradhan, M. N. Jebur, H. Z. M. Shafri, and M. S. Tehrany, "Data fusion technique using wavelet transform and taguchi methods for automatic landslide detection from airborne laser scanning data and quickbird satellite imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1610–1622, Mar. 2016, ISSN: 01962892. DOI: 10.1109/TGRS.2015.2484325.

[60] G. Sohn and I. Dowman, "Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 1, pp. 43–63, May 2007, ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2007.01.001.

[61] S. E. Franklin and C. F. Blodgett, "An example of satellite multisensor data fusion", *Computers and Geosciences*, vol. 19, no. 4, pp. 577–583, Apr. 1993, ISSN: 00983004. DOI: 10.1016/0098-3004(93)90083-H.

[62] J. Verbesselt, A. Zeileis, and M. Herold, "Near real-time disturbance detection using satellite image time series", *Remote Sensing of Environment*, vol. 123, pp. 98–108, 2012, ISSN: 00344257. DOI: 10.1016/j.rse.2012.02.022.

[63] K. R. McCloy and W. Lucht, "Comparative evaluation of seasonal patterns in long time series of satellite image data and simulations of a global vegetation model", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 1, pp. 140–153, Jan. 2004, ISSN: 01962892. DOI: 10.1109/TGRS.2003.817811.

[64] P. Coppo, L. Chiarantini, and L. Alparone, "End-to-End image simulator for optical imaging systems: Equations and simulation examples", *Advances in Optical Technologies*, 2013, ISSN: 16876393. DOI: 10.1155/2013/295950.

[65] S. Borra, R. Thanki, and N. Dey, "Satellite Image Classification", in *SpringerBriefs in Applied Sciences and Technology*, Springer Verlag, 2019, pp. 53–81. DOI: 10.1007/978-981-13-6424-2_4. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-13-6424-2%7B%5C_%7D4.

[66] M. Pal and P. M. Mather, "Support vector machines for classification in remote sensing", *International Journal of Remote Sensing*, vol. 26, no. 5, pp. 1007–1011, Mar. 2005, ISSN: 01431161. DOI: 10.1080/01431160512331314083.